2017-09

Proof of concept in disrupted tactical networking

Kline, Thomas D.

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

PROOF OF CONCEPT IN DISRUPTED TACTICAL NETWORKING

by

Thomas D. Kline

September 2017

Thesis Advisor: Alex Bordetsky
Second Reader: Steve Mullins

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2017	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE PROOF OF CONCEPT IN DISRUPTED TACTICAL NETWORKING	**5. FUNDING NUMBERS**
6. AUTHOR(S) Thomas D. Kline	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Current systems used to control unmanned assets and maintain command and control networks typically rely upon persistent signals. However, the Department of Defense (DoD) predicts that adversaries will be able to detect, geolocate, and target through electromagnetic (EM) spectrum operations in the future operating environment. Unable to rely upon constant interconnection, the DOD must begin to reconsider the nature and behavior of its networks. In 2011, Bordetsky and Netzer proposed "networks that do not exist" as a potential solution. They envision multi-domain networks whose links connect only long enough to transmit critical information securely. The links quickly disconnect, leaving no trace electromagnetically.

The DoD lacks sufficient research that evaluates the merits of short-living network solutions. Without adequate research, the future DOD may either unnecessarily expose its forces to adversaries through the networks or impair decision-making by choosing not to communicate because of the risk of detection. In this study, we design projectile-based mesh networking prototypes as one potential type of short-living network node and use the projectiles to observe some of the merits and challenges of moving from persistent signal networks to cluster-based networks created only during disruption.

14. SUBJECT TERMS mesh networking, projectile-based, bursty tactical networks	**15. NUMBER OF PAGES** 161
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01-280-5500 Standard Form 298 (Rev. 2–89) Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

PROOF OF CONCEPT IN DISRUPTED TACTICAL NETWORKING

Thomas D. Kline
Major, United States Marine Corps
B.A., Tulane University, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2017

Approved by: Alex Bordetsky
Thesis Advisor

Steve Mullins
Second Reader

Dan Boger
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Current systems used to control unmanned assets and maintain command and control networks typically rely upon persistent signals. However, the Department of Defense (DoD) predicts that adversaries will be able to detect, geolocate, and target through electromagnetic (EM) spectrum operations in the future operating environment. Unable to rely upon constant interconnection, the DOD must begin to reconsider the nature and behavior of its networks. In 2011, Bordetsky and Netzer proposed "networks that do not exist" as a potential solution. They envision multi-domain networks whose links connect only long enough to transmit critical information securely. The links quickly disconnect, leaving no trace electromagnetically.

The DoD lacks sufficient research that evaluates the merits of short-living network solutions. Without adequate research, the future DOD may either unnecessarily expose its forces to adversaries through the networks or impair decision-making by choosing not to communicate because of the risk of detection. In this study, we design projectile-based mesh networking prototypes as one potential type of short-living network node and use the projectiles to observe some of the merits and challenges of moving from persistent signal networks to cluster-based networks created only during disruption.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

A2/AD	Anti-Access/Area Denial
ACK	Acknowledge
AO	Area of Operations
AOA	Angle of Arrival
AODV	Ad Hoc Distance Vector
API	Application Program Interface
ARG	Amphibious Ready Group
C2	Command and Control
CA	Collision Avoidance
CCIR	Commander's Critical Intelligence Requirement
CD	Collison Detection
CDM	Code Division Multiplexing
CENETIX	Center for Network Innovation and Experimentation
CIA	Confidentiality, Integrity, and Availability
CONOPS	Concept of Operations
COP	Common Operational Picture
COT	Cursor On Target
COTS	Commercial Off The Shelf
CTS	Clear To Send
CSMA	Carrier Sensing Multiple Access
DARPA	Defense Advanced Research Projects Agency
DOD	Department of Defense
DON	Department of the Navy
DTN	Disruption Tolerant Network
DTNRG	Disruption Tolerant Network Research Group
EM	Electromagnetic
EMCON	Emissions Control
EMS	Electromagnetic Spectrum
EMW	Electromagnetic Warfare
FCC	Federal Communications Commission
FDM	Frequency Division Multiplexing

FHSS	Frequency Hopping Spread Spectrum
FIPS	Federal Information Processing Standards
FISMA	Federal Information System Management Act
GEO	Geosynchronous Earth Orbit
IEEE	Institute of Electrical and Electronics Engineers
IETF	International Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IPV4	Internet Protocol Version 4
IPSEC	Internet Protocol Security
IR	Intelligence Requirement
ISM	Industrial, Scientific, Medical
ISO	International Standards Organization
ISR	Intelligence, Surveillance, and Reconnaissance
LAN	Local Area Network
LEO	Low Earth Orbit
LCS	Littoral Combat Ship
LFOC	Landing Force Operations Center
LLC	Logical Link Control
LOS	Line of Sight
MAC	Media Access Control
MANET	Mobile Ad Hoc Network
MEU	Marine Expeditionary Unit
MHZ	Mega Hertz
MSS	Maximum Segment Size
NASA	National Aeronautics and Space Administration
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NPS	Naval Postgraduate School
OLSR	Optimized Link State Routing
OS	Operating System
OSI	Open Systems Interconnect
PIR	Priority Intelligence Requirement
PLT	Pneumatic Line Thrower

RFC	Request For Comment
RREQ	Route Request
RREP	Route Reply
RRER	Route Error
RSS	Received Signal Strength
RTS	Request To Send
SNR	Signal to Noise Ratio
SP	Special Publication
SSL	Secure Socket Layer
SYN	Synchronize
TCP	Transport Control Protocol
TDM	Time Division Multiplexing
TDMA	Time Division Multiple Access
TLS	Top Layer Security
TOA	Time of Arrival
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UGV	Unmanned Ground Vehicle
UUV	Unmanned Underwater Vehicle
UXS	Unmanned System
VE	Virtual Extension
VEmesh	Virtual Extension Mesh 290S
VPN	Virtual Private Network
WMN	Wireless Mesh Network

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First, I would like to thank my wife, Niki Kline, for supporting me through this adventure. Your amazing support is what allows me to strive for success and your love is the bed rock of our family. Thank you for everything, honey.

I would like to thank the CENETIX team. Dr. Bordetsky, you have empowered me at every turn and I admire your unique vision for what is in the realm of the possible. Steve Mullins, thank you for your timely advice and perspective. As the utility player of the CENETIX team, I admire the way that your efforts tie the team together. Eugene Bourakov, you are a tremendous asset to the Naval Postgraduate School. Thanks for lending me your expertise and your time. I can see why Dr. Bordetsky has developed a belief that anything is possible. You have a knack for making it so. Malcolm Mejia, thank you for showing me the ropes of mesh networking and for your friendship. I wish everyone on the CENETIX team continued success, and will remain willing to help in any way that I can.

Finally, I would like to thank Kristen Tsolis and the staff that volunteer at the Robo-Dojo. The Naval Postgraduate School's Robo-Dojo generously provided their facility for our prototyping process efforts. The various seminars hosted there had the effect of lowering my perceived cost of entry into the field of robotics, programming, and manufacturing. As a creative space for students and faculty, the Robo-Dojo makes it easy to learn a number of skills including 3D printing, laser cutting and etching, computer programming, and micro-electronics. It is such a valued resource to the student body, and I hope that it continues to grow.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Current communication systems for controlling unmanned systems (UXS) and maintaining command and control typically rely upon robust and persistent signals. However, recent developments in cyberspace and electromagnetic (EM) spectrum operations promise to challenge the Department of Defense's (DOD) reliance upon persistent connections in time and space (Department of the Navy [DON], 2015). The recently published Marine Corps Operating Concept states the situation in the battle of signatures section: "[t]omorrow's fights will involve conditions in which "to be detected is to be targeted is to be killed" (United States Marine Corps [USMC], 2016a, p. 10).

Advances in EM geolocation technology threaten to spread the ability to target forces by their EM signals from strictly high-end adversaries to a much wider array of threats. This means that communicating and controlling through persistent signals could soon be the operational equivalent of setting up an ambush only to have a team member stand up and loudly announce all friendly positions. Operating persistent data networks also allows adversaries to continuously observe DOD forces and our coalition partner networks, providing greater opportunity to discover our vulnerabilities. As EM and cyber warfare become prominent warfighting considerations, technological maturation in fields such as autonomy may allow the DOD to reconsider the fundamental qualities of its networks. Autonomous assets promise to function without the need for persistent connections. Significant research continues for the purpose of discovering the possibilities of operating autonomous systems. Singer's popular book, *Wired for War* (2009), presents a few of those possibilities. Examining how information might flow when autonomous systems are operational is an interesting sub-task of that research. The autonomous system information flow question feeds nicely into research that continues in mesh networking.

In 2011, Bordetsky and Netzer proposed "networks that do not exist" as a potential solution. They envision multi-domain networks whose links connect only long enough to transmit critical information securely. The links quickly disconnect, leaving no trace electromagnetically. Bordetsky and Netzer labeled disruption-based or bursty

tactical networks. In Bordetsky and Netzer's hypothesized network, nodes would advertise, authenticate, determine routes, transmit critical data, acknowledge receipt, and disconnect—all before an adversary is able detect that the network exists. The receiving mesh nodes would store the critical data and then simply wait for another burst, or they might physically travel within range of other nodes to start a new burst at a different discrete moment in time, potentially at a location far-removed from the first burst.

The possible applications for such a network are very interesting to consider. Bordetsky, Benson, and Hughes (2016) conceptualize that "hard to detect-hard to compromise" nodes could support the Littoral Combat Ship's (LCS) new operational roles in the littoral combat area. In an example case proposed by Bordetsky (2016), the LCS links to autonomous data-collection systems by shooting projectiles with mesh networking payloads embedded. The payloads communicate in burst transmissions between the ship, fast patrol boats, and unmanned assets, perhaps during moments of cube satellite orbital node availability, "all in a coordinated dance" (Englehorn, 2017). In less than eight seconds, the projectiles are destroyed, leaving an adversary with only vague knowledge that an event had occurred. The disruption-based networking approach Bordetsky proposes in littoral combat is an interesting case. What other tactical or operational applications might also be a fit? Where would disruption-based networks be inappropriate? What is the best way to employ a disruption-based network? What are desirable features of nodes within such a network? What type of features are desirable for routing, authentication, and data transfer protocols?

The problem is that very little research has been done to evaluate the merits and challenges of operating disruption-based networks. There are very few available networking devices that are designed to be short living and highly mobile. Bordetsky's hypothesized projectile does not yet exist commercially. Without sufficient examination, the future DOD could either unnecessarily expose its forces to EM detection and targeting by communicating persistently or conversely induce poor decision-making by deciding not to communicate in light of the adversarial threat.

This thesis represents a beginning for disruption-based networking research using short-living and highly mobile nodes. It is presented as a proof of concept. Our initial

efforts to locate and acquire short-living, highly mobile nodes for testing were not successful. Fortunately, maturing mesh networking technologies and the reduced size and cost of computing assets make it possible to create expendable, short-living network nodes of our own. In this thesis, we design projectile-based mesh networking prototypes and experiment with them. Projectiles are simply one potential type of short-living network node. There are many other types that may also work. However, we use projectiles to observe some of the merits and challenges of moving from persistent signal networks to cluster-based networks established only by disruption. Projectiles by their nature are short-living. In this research, we limit our communications window to the duration of the projectile's flight. In summary, we examine information flow in a network organized to be discretionary in time and space for the purpose of examining the feasibility of bursty-tactical networks.

A. RESEARCH OBJECTIVES

This study examines information patterns in a network that has been organized in a fundamentally different way—discretionary in time and space. We observe and record some of the challenges of moving from persistent signal networks to cluster-based networks interconnected only by disruption.

The primary question addressed in this research is the following:

How does information flow in networks that are interconnected only by disruption?

By observing some of the challenges of operating short-living nodes, this research ultimately collects insights about desirable operating features of the nodes themselves. Because we prototype nodes for a proof of concept, we also inform corollary objectives that include exploring the coupling of additive manufacturing with low-cost technologies, and potential fits for tactical employment of projectile-based nodes.

B. SCOPE AND LIMITATIONS

The research question explores information flow in a disruption-based network prototype designed as a way to support command and control in EM-hostile

environments. It focuses on observing the behaviors of the network and the nodes in order to postulate about desirable features of the nodes and their interactions. This thesis remains purposefully in the unclassified domain, which significantly limits its scope. However, it preserves the opportunity to reach a broader DOD audience with the hope to inspire follow-on work.

We do not use electromagnetic detection tools in order to attempt to geo-locate our experimental nodes during the duration of their interactions. That is beyond the scope of this work; rather, we focus on the selection of components and prototyping projectile-based nodes. This work is accomplished through basic experimentation, using existing and inexpensive commercial-off-the-shelf (COTS) components, and does not attempt to modify manufacturer-set protocols in order to optimize results.

C. ORGANIZATION OF THESIS

Chapter II contains a literature review of relevant supporting research, theory, and concepts. Chapter III describes the research design and experimental modeling conducted to demonstrate the possibility of operating with network nodes that communicate only by disruption. Chapter IV provides the necessary technical background for the reader to understand the experimentation results. Chapter V recounts the prototyping process and provides observations and conclusions. Chapter VI summarizes the significant findings and provides recommendations for future work.

II. LITERATURE REVIEW

This chapter is divided into three main sections. The first section reviews networking in the future operating environment. The future operating environment is a significant driver of research in operating communication networks outside of persistent signal architectures. The second section addresses the current state of our knowledge of networks that depart from persistent signal architectures. The first and second sections are intended to frame the basis for the proposal for researching bursty-networking nodes.

The final section reviews current network science research for networks that display bursty behavior. Although the networks discussed in the final section expand beyond telecommunication-type networks, they are a vital part of the total body of knowledge pertaining to network behavior in conditions when connections are not persistent.

A. NETWORKING IN THE FUTURE OPERATING ENVIRONMENT

The DOD's position is that there is a growing need to evolve the way command and control is exercised. According to Joint Publication 3-0,

> the electromagnetic spectrum, which has become increasingly complex, contested, and congested as technology has advanced, can significantly affect joint force operations. Operational experiences demonstrate not only how successful control of the EMS can influence the outcome of the conflict, but highlight U.S. dependence on the EMS in order to successfully operate. (2011, p.V-43)

In order to illustrate the growing need for alternative networking solutions, consider the Navy and the Marine Corps perspectives in the following sections.

1. Naval Combat in the Littorals

Operating unopposed since the end of the cold war, the Navy has become used to freely sharing information (Angevine, 2011). The Navy developed robust systems that equip its commanders with unprecedented tools with which to command and control naval forces. Ironically, it is the Navy's unprecedented tools that add risk to operating in

the littorals. Current command and control networks emit a significant and detectable EM signature. The Navy expects adversaries to use emerging electromagnetic spectrum (EMS) technologies to detect, identify, and triangulate locations in the littorals (DON, 2015). As EM tools emerge and are coupled with mines and anti-ship ballistic missiles, they create additional anti-access/area denial (A2AD) challenges for the Navy.

Regardless of the threat, the Navy has a mandate to project power in the littorals, which is codified in the Department of the Navy's (DON) Cooperative Strategy for 21st Century Seapower (2015). The DON's strategy document confirms that: "[n]ew challenges in cyberspace and the [EM] spectrum mean [that the Navy] can no longer presume to hold the information 'high ground.' Opponents seek to deny, disrupt, disable, or cause damage to [naval] forces… with advanced networked information systems" (DON, 2015, p. 8). In order to respond to rising A2/AD challenges, DON states that the Navy will "develop a force capable of effective, autonomous operations in an information-denied or -degraded environment" (p. 33, 2015). Operating in information-degraded environments is not a new naval concept. Key tenets of command and control in the maritime domain have long been "the necessity of the subordinate commanders to execute operations independently … with a thorough understanding of the commander's intent, and command by negation" (Joint Chiefs of Staff, 2013, p. I-2).

According to the Navy, no matter how hostile the EM conditions are, elements of a modern naval force must share information (DON, 2015). Commanders must receive critical information in order to maintain situational awareness and make informed decisions (Joint Publication 3-0, 2011). Elements of any naval or joint force need to maintain a common operational picture (COP) in order to work together effectively.

The Navy created its electromagnetic maneuver warfare (EMW) concept to achieve EM resilience, attempting to disable the adversary's A2/AD targeting capabilities through the use of cyberspace and the EM spectrum (DON, 2015). However, the Navy should not assume that its EMW will succeed when needed and should not continue to rely solely on persistent networks. When disabling adversarial targeting capabilities fails, the Navy will return to its reliance upon emission control (EMCON) to achieve EM resilience (Angevine, 2011). The idea of EMCON is that by disconnecting persistent

networks, a naval force reduces the adversary's ability to detect the presence of and target friendly naval forces. However, the surface Navy no longer trains to operate while disconnected. The Navy is currently investigating ways to command and control (C2) the naval force in any EM hostile environment (DON, 2013). Disruption-based networking fits well within the Navy's search of C2 innovations.

2. Marine Corps Future Operating Environment

The impetus for the *Marine Corps Operating Concept* published in 2016 is the Marine Corps' own recognition that it "is not organized, trained, and equipped to meet the demands of a future operating environment characterized by complex terrain, technology proliferation, information warfare, the need to shield and exploit signatures, and an increasingly non-permissive maritime domain" (USMC, 2016a, p. 12).

Listed as a critical task, the Marine Corps states that it must:

> exhaust all possibilities to protect our C2 and information networks while simultaneously exploiting networking to put ourselves into position to gain all the possible advantages thereof. This includes operating with ruthless prioritization of information sharing between the various command echelons while being prepared to operate with imperfect information. We must take into account the role of signature in offense and defense to mitigate the enemy's targeting of our network and exploit enemy C2 vulnerabilities. We must shorten the kill chain by networking for rapid/precise fires and pushing processing power to the tactical edge. (USMC, 2016b, p. 6)

Marine Corps leadership envisions a future where Marines fight in complex, urban areas that "are the most likely to occur and the most dangerous. (2016a, p. 25)" Urban terrain is complex, both geographically and in the EM spectrum. Marine Corps leadership specifies initiatives in manned-unmanned teaming as well as shortening the "kill chain" by closely linking geographically distributed forces with intelligence, surveillance, and reconnaissance (ISR) sensors and fires. In a scenario that places those distributed teams in hostile EM conditions, the need for alternatives to the persistent networking model becomes urgent. Emerging technologies make those hostile EM conditions more likely in the future operating environment.

3. Maturing Technological Threats

There are emerging technologies that Navy and Marine Corps leaders believe could threaten friendly persistent networks. Two of these threats include the proliferation of commercially available EM detection and targeting tools and near-peer adversary weapon development.

a. Targeting Space

Satellite communication systems have long been a means to broadcast information to forces outside of the terrestrial communications infrastructure, which is where the DOD frequently operates. Likewise, for a long time, space was seen as a sanctuary (Deblois, 1998). That perception began changing in 2007, when China successfully destroyed one of their antiquated weather satellites with a SC-19 direct-ascent weapon (Kan, 2007). According to Lewis in a Foreign Policy magazine article (2014), China tested their antisatellite (ASAT) four times. China's current ability to target objects in low earth orbit (LEO) is a clear demonstration that the DOD's ISR satellite networks are threatened. It is only a matter of time before networks using geosynchronous earth orbits (GEO) and other orbits are also in jeopardy. The implications of losing such persistent communication systems raises many questions. How would the DOD command and control its forces without satellite communications?

b. Geolocation Tools Become Available

Sayed, Tagrihat, and Khajehnouri (2005) detail how in 1996, the Federal Communications Commission (FCC) mandated that wireless carriers report locations of users who place 911 calls using wireless devices. That mandate spurred significant enhancements of wireless location algorithms. Sayed et al. also cite other motivations that will increase radiodirection technology, most interestingly the function of mobile advertising. Mobile advertisers hope to be able to offer companies with the ability to generate just-in-time ads. Think of a case where a driver is listening to Internet radio when an add plays about a restaurant just up ahead, or the driver approaches a billboard which has changed to advertise something that predicts the driver's needs based on his or her browsing history. Patwari et al. (2005) describe measurement-based statistical models

that use time-of-arrival (TOA), angle-of-arrival (AOA), and received-signal-strength (RSS) that may reappear in geolocation algorithms used in the littoral areas for targeting purposes. Drones that carry EM geolocation equipment exist, and according to Gruss (2013), the market for more unmanned systems using EM detection and location tools exists as well. As EM geolocation tools proliferate, the DOD cannot ignore the probability that adversaries of all types will attempt to use them to enhance their targeting capabilities.

B. CURRENT RESEARCH IN NON-PERSISTENT SIGNAL COMMUNICATIONS

The previous section detailed some of the motivations behind disruption-based network research. This section provides an overview of research efforts that parallel the focus of this thesis. Prior work with data communication networks modeled as clusters connected only by disruption, or bursts, is limited. However, there is an interesting line of research that focuses on successful transfer of data when the entire path from sender to destination cannot be achieved synchronously (Sehl, 2013). There is also a proposal to use projectile-based networking nodes as a control channel to provide waypoint instructions in tactical mesh networks (Bordetsky and Netzer, 2010). There is also a projectile that transmits video images back to the shooter during flight (Rafael, n.d.). These research efforts will be addressed individually in the following sections.

1. Delay and Disruption-Tolerant Networking (DTN)

Delay and disruption-tolerant networks (DTN) are a relatively new networking architecture conceived for environments where an end-to-end path from sender to receiver may not be possible. Delay tolerant networks generally refer to networks which have to overcome long latency due to distances and availability (Sehl, 2013). Disruption tolerant networks, on the other hand, generally refer to a wider range of obstacles to overcome. These obstacles include myriad issues such as transmission distances and intentional attack (Sehl, 2013). The main driver of DTN research was the interplanetary Internet, conceived for use in interplanetary communications and deep space exploration (Cerf et al., 2007). The National Aeronautics and Space Administration (NASA) did not

use an Internet-style architecture to communicate with satellites and spacecraft in the past. Instead, NASA has used point-to-point or single relay LEO links to communicate with spacecraft (National Aeronautics and Space Administration [NASA], n.d.). However, NASA predicts that future exploration will not be successful using the point-to-point communication model. As exploration goes deeper into space, point-to-point connection opportunities will be of limited duration and latency will hinder successful data transfer. NASA believes that overcoming these more complex environments will require data transfer between many nodes. NASA believes they need a communication model akin to the Internet only in space. However, traditional earth-bound Internet functionality seems ill-suited to scale up for the space environment. Figure 1 illustrates delay- and disruption-tolerant networking architecture.

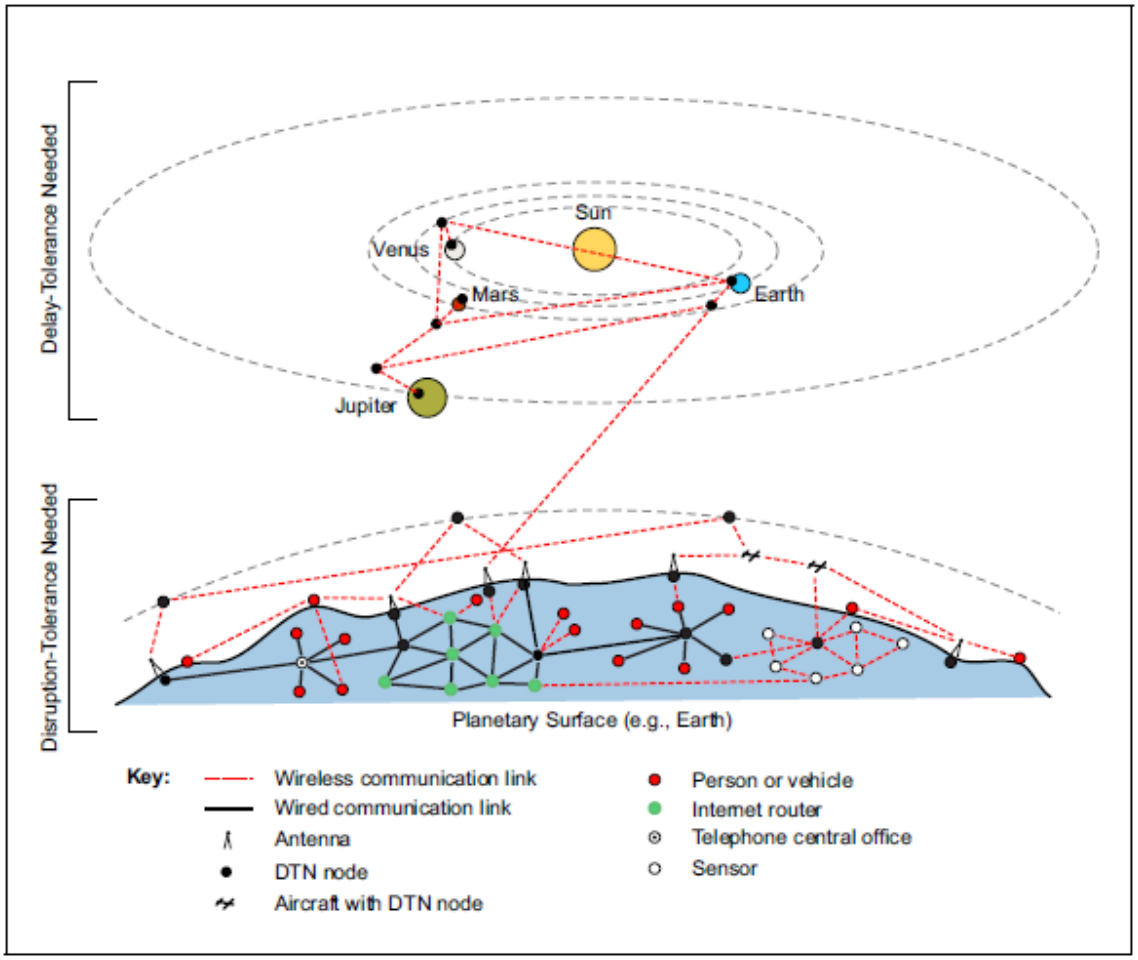

Figure 1. Delay- and Disruption-Tolerant Network. Source: Warthman (2015).

The Internet architecture on earth is ill-suited to scale up for the space environment because it was and is developed with a few key assumptions that do not hold true in the space environment. Cerf et al. (2007) list several of these fundamental assumptions in their Delay Tolerant Networking Architecture request for comment (RFC). Cerf et al. note the assumptions that:

- an end-to-end path exists between source and destination for the duration of a communication session
- retransmission based on timely feedback from the receiver is an effective method to repairing errors
- end-to-end loss is relatively small

These assumptions led to the development of protocols designed to treat the two nodes as a back-and-forth conversation style communication. These protocols range from reliable transport of information to encryption for confidentiality. Protocol developers optimized functionality for the terrestrial environment. The problem is that in the space environment, an end-to-end path may not exist. Orbiting planets and satellites have windows in which connections are possible and windows where connections are not. In fact, many other underlying assumptions in the Internet also do not hold true in space. Space environments also experience long delays due to sheer distances and high error rates due to radiation and other factors. Thus, traditional inter-networking protocols used in space produce errors, significant delays, poor performance, and failure.

The general idea of disruption-tolerant networks is that the routers along the path from sender to destination use a store-and-forward model instead of a simple route-and-relay model. The DTN-capable nodes store bundles of data until such a time when connection to the next node is possible. When connection with the next node is made, the DTN-capable nodes forward their bundles. Think of delay-tolerant networking as the pony express model of communications. The pony express comprised a series of stops where riders would exchange bundles and carry them to the next stop before going back for more. DTN, like the pony express, has a custody exchange feature (Cerf et al., 2007). Detailed description of both traditional Internet architecture and the disruption-tolerant network architecture is provided in Chapter III.

Delay/Disruption tolerant networking architecture appears to be quickly maturing. DTN was proved as a concept in 2002 and NASA began working with the Defense Advanced Research Projects Agency (DARPA) in 2004 on the next generation DTN (NASA, n.d.). The network working group at Internet Engineering Task Force (IETF) published two RFCs in 2007. NASA deployed the first DTN capability to the international space station in 2014. Interestingly, the Delay Tolerant Networking Research Group (DTNRG) believes that the DTN architecture will fit naturally in several environments other than space. In RFC 4838, the DTNRG specifically recommends DTN in sensor-based networks using scheduled intermittent connectivity, satellite networks with periodic connectivity windows, underwater acoustic networks, and terrestrial

wireless networks that cannot maintain end-to end connectivity (Cerf et al, 2007). The DOD has noticed DTN's potential application in tactical networking as well. Sehl (2013) examines a Raytheon-produced DTN software product for suitability for use by the United States Marine Corps. DTN fits as a major consideration in this thesis' proposed multi-domain, tactical mesh network communicating by disruption only because of EM detection probability. A few of these considerations are detailed in following chapters.

2. Projectile-Based Nodes

Bordetsky and Netzer (2010) describe the Naval Postgraduate School's (NPS) Center for Network Innovation and Experimentation (CENETIX) and mesh network projects intended to further the development of interagency collaboration. In their report, Bordetsky and Netzer list projectile-based mesh networks as a potential area of experimentation. Their idea evolved logically as a potential method to transmit waypoint management information in autonomous unmanned aerial systems (UAS). Bordetsky further describes a potential role for projectile-based nodes in *Patterns of Tactical Networks* (2012). In describing the future of manned-unmanned teams, Bordetsky describes how decentralized computation services, made possible by emerging technologies, can reduce the amount of data required to be transmitted to a central information system designed to provide both operational decision support and network management (2012). According to Bordetsky, "ongoing field experimentation with tactical networking environments clearly indicates that disruption-based networking could become one of the major trends in the emerging tactical services" (2012, p. 8). Bordetsky hypothesizes that tactical sensor networks, unmanned systems, and moving operators could combine to negate the typical requirement to maintain wireless connections in any network. Bordetsky describes "a disruption-based model of networking at high-speed… in which two-way communication takes place during 2–8 seconds of the grenade type device slowed down descent to the area of interest" (2012, p. 8). Bordetsky specifically names the Firefly as a prototype for testing such short-burst tactical networks (2012). The Firefly is discussed in the next section.

3. Rafael Firefly

Rafael (n.d.) envisioned the Firefly as an enabler for a squad-size unit in an urban combat environment. According to Rafael, the Firefly is a 40mm grenade-launched video camera, sending imagery back to a tactical unit without the need for line of sight. Rafael advertised the Firefly's key features as streaming video, high-resolution photo, and quiet launch. The Firefly's maximum range is 600m, and maximum apogee is 150m. At maximum altitude, the Firefly's resolution is 20cm per pixel. Although not a two-way data communication system, the Firefly is a proof of concept that data can be transferred from a small projectile back to the grenade launcher and tablet. Figure 2 shows the Firefly and its components:

Figure 2. Firefly and components. Source: Rafael, (n.d.).

We attempted to obtain a Firefly, but it was no longer in production by or supported from the Rafael Corportation. We thereupon decided to conduct a feasibility analysis for prototyping our own using commercially-available electronic components and our own 3D-printed assembly designs.

C. BURSTY NETWORKS

This section reviews current network science research for networks that display bursty behavior. The volume of previous research in telecommunication networks modeled as clusters connected only by disruption, or bursts, is finite. However, bursty behavior within various network types is well-studied under the network science field. Barabasi is a respected voice in network science, and Barabasi's work (2010) offers some context about naturally occurring short-lived networks that is useful for this study. While *networks of wealth* examples are seemingly unrelated to the study of data communication networks, Barabasi's study finds bursty patterns in systems where randomness would initially appear more likely. Barabasi's findings are of particular interest because they offer the possibility that communications over the DOD's command and control systems may also exhibit burstiness, where previously the frequency of those communications might have been assumed as random. The possibility that application layer command and control data exhibits bursty characteristics would suggest that the projectile and short-living nodes are actually well-suited for use in tactical networks. The following sections detail Barabasi's contributions to the study of network patterns. Although not tied directly into telecommunications, the concepts are relevant and important in the context of this study.

1. Bursty Versus Random Network Patterns

In *Bursts* (2010), Barabasi explains that most of the technologies of modern life are the result of hundreds of years of scientific inquiry endeavored with the enduring belief that, even though they were yet unknown, there are laws that explain natural phenomena. *Bursts* is the chronicle of Barabasi's pursuit to discover those laws by examining human systems for patterns and then developing models to recreate those patterns. While *Bursts* is not specific to data networks, Barabasi's insights enrich the meaning of this study.

Barabasi (p.85, 2010) begins by pointing out the unexpected patterns produced by truly random systems. He offers the common example of a random system: rolling dice. Rolling a die is a truly random outcome; the chances of resting on the side facing up is

one in six. Barabasi shows a sequence of 400 roles, marking a dot for each roll that results in one through five, and a slash for every six that is rolled. Figure 3 shows the results of Barabasi's die roll sequence.

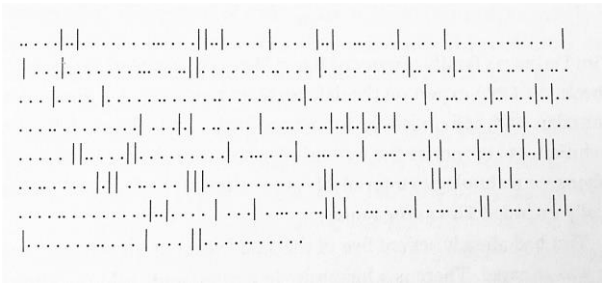

Figure 3. Die Roll Sequence. Source: Barabasi (2010).

The sequence appears random but at the same time rather uniform. Most of Barabasi's sixes appear within five to seven rolls of each other. Barabasi points out that the die roll system is not likely to go 10 straight rolls each resulting in a six nor would it be probable to get a remarkably long string absent of sixes. The system would have to roll one hundred million times for that event to occur probabilistically. Barabasi uses the example system to demonstrate that truly random systems result in Poisson distributions.

Barabasi proposes that many human systems also demonstrate patterns following the Poisson distribution. Barabasi (pp. 98–102, 2010) cites Richardson's *Statistics of Deadly Quarrels* (1950) as a direct example. Richardson catalogues conflicts and wars that occurred between 1820 and 1949 in an attempt to find causal factors. Barabasi notes that Richardson found no causal factors in the data: they appeared random. According to Poisson, if wars are truly random then they should each have roughly the same number of casualties. However, the amount of casualties varied greatly. Richardson assigned a base 10 logarithmic scale according to the amount of deaths in each war, giving a magnitude zero value to conflicts with few casualties and a magnitude seven value to the wars that took millions of lives. Barabasi describes Richardson's findings as "the fewer, the larger" (p. 102). World Wars I and II are the lone magnitude sevens, while 188 of the 282 other wars were of magnitude three or less. Barabasi also cites the more famous economist Pareto and Pareto's work in networks of wealth. Pareto discovered that while the vast

majority of people are poor, a select few garner vast wealth (p. 102, 2010). Pareto's work became known as Pareto's law, where in many systems 20% of the independent variables collect 80% of the dependent variables.

Barabasi also studies the pattern of email traffic, first citing personal email data and then using a larger data set provided by Eckman (pp. 101–103, 2010). Interestingly, Barabasi notes that 80% of email is sent in 20% of the time. Email networks follow the Pareto law. Plotted against time, emails are absent altogether for long periods of time, then are sent in bursts. Barabasi finds that phone calls are made in bursts too. Barabasi's opinion is that the reason email burstiness fascinates is "precisely because it is not unique to our email pattern" (p. 104).

Barabasi's perception of ubiquitous burstiness begs questions whether those patterns carry over to the military: might command and control communications in a tactical force also exhibit burstiness? There are some initial hints that the prospect is a worthy inquiry. It is a common observation by many who have been in combat recalling long periods of inactivity followed by short periods of frantic action. If command and control communications during missions prove to exhibit bursty patterns, might a bursty network—one that exists only by disruption—adequately support command and control requirements?

Since bursty tactical networks are only theoretical, the first step to address these questions is to prove that networking in such a way is possible. This thesis is designed as a proof of concept, prototyping projectile-based nodes and experimenting with their use. Selecting the components of the projectile-based node and understanding of the network behavior in our experiments requires technical understanding of data communication network technologies and protocols. The next chapter provides this technical background.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TECHNICAL BACKGROUND

This chapter provides the technical background of the thesis. The technical background is provided as three main reference frameworks. The first section outlines the open systems interconnect (OSI) framework. This study considers each layer in the OSI model while examining network and EM signature behavior. The second section contains the confidentiality, integrity, and availability (CIA) framework model. The CIA model provides relevant considerations for the desired behavior of tactical networks. The final section provides the reader with a background in systems-theory. This study is designed using systems theory and uses systems theory as a lens through which to observe network behavior. And finally, the third section provides a framework for the reader to understand the technologies and protocols that this study observes in order to explain node and network behavior during the experiments. The third section will assist the reader in understanding our findings and recommendations.

A. OPEN SYSTEMS INTERCONNECT (OSI) MODEL

This section discusses the OSI model in layman's terms. Readers who are familiar with these frameworks should skip this section.

At the most basic level, all communication involves an entity that sends information and an entity that receives it (Comer, 2015). Comer states that those two entities must agree on several things for communication to be possible. The range and number of these agreements may not seem obvious. To illustrate a few of the necessary agreements, consider an example of when one person, Alice, wishes to communicate with another person, Bob. Alice typically makes a choice to use voice, making the assumption that Bob will be both able to hear her and also understand her language. Alice first detects that Bob is not otherwise engaged, and begins by saying "Hello, Bob" or by making good eye contact. If Alice interrupts Bob's existing conversation, she would be breaking etiquette and her interruption would likely interfere with the existing conversation. If Alice began the conversation without ensuring Bob knows that Alice is talking to him, Bob would probably miss some information and Alice would have to start

over. Beginning conversations without interruption, gaining attention, and beginning with ‘hello’ are all agreements that the reader may more easily recognize as etiquette. If Alice is writing, she uses commas and periods to frame her thoughts and to indicate to Bob that pausing is necessary. Bob would find it difficult to follow Alice’s writing if she omits those natural breaks. Data also has breaks, characterized at the most basic level as *frames*. In data communications, these etiquette agreements are called protocols. A protocol is simply a set of steps that need to be followed. These agreements, whether etiquette or protocol, are designed to ensure successful communications. These agreements are also of varying degrees of complexity. The OSI model provides a common frame of reference for agreements of different complexity. The OSI model divides protocols by the function they perform (Comer, 2015). The OSI model is an essential frame of reference for this study. Specifically, this study seeks to understand how the protocols in the experiments affect network behavior. Through this analysis, this study provides insight about favorable characteristics of protocols resident in nodes within a disruption-based network.

The OSI model’s divisions are commonly called ‘layers’ and each layer is ‘stacked’ by the sophistication of the function that the layer performs. The OSI model has seven layers as depicted in Figure 4.

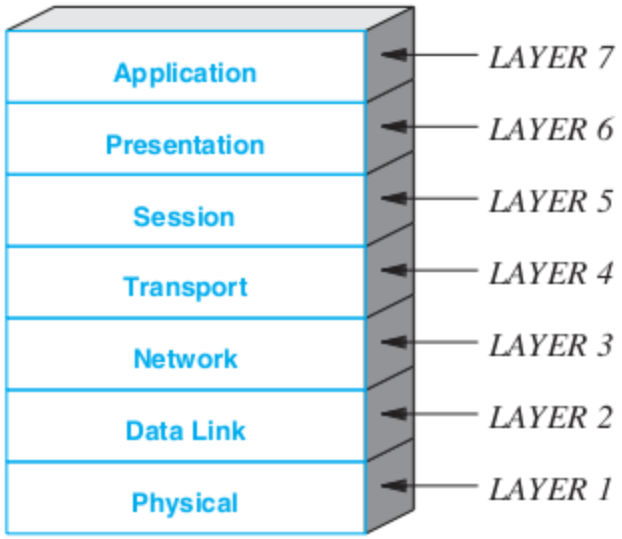

Figure 4. OSI 7-Layer Communication Model. Source: Comer (2015).

The layered approach allows for developing new ways to perform a specific function without requiring developers to modify each of the other functions as well. A full technical description of the OSI model is in the organization for international standardization (ISO)/IEC 7498–1 (1996). The following sections address the each of the seven layers and provide common examples in both fixed and mobile network types.

1. Physical Layer

The physical layer is the lowest logical division in the OSI stack and deals primarily with the mediums themselves, which are classified as guided or unguided. Examples of physical layer mediums include Ethernet cable and IEEE 802.11 (Wi-Fi). Physical layer properties are important in this study. Functions at all higher layers are translated into a physical layer signal. Adversary detection of the physical layer's signal is a driving force to study alternative networking methods and is a significant consideration throughout this experimentation with disruption-based networking.

This research uses unguided, or wireless, mediums. Wireless mediums are suitable in tactical networks where nodes are mobile. Agreements made at the physical layer are agreements about the physical properties of the signals themselves. Any two

nodes must agree on the medium used, the bandwidth (defined simply as the highest frequency to the lowest frequency), and the data encoding technique.

Data encoding is performed by modulation in analog carriers and by shift keying in digital signals (Comer, 2015). Common modulation techniques include amplitude modulation and frequency modulation (Comer, 2015). A signal using amplitude modulation keeps the frequency constant while varying the amplitude according to the data being carried. A signal using frequency modulation keeps the amplitude constant while varying the frequency. Figure 5 illustrates a carrier wave with amplitude modulation on the left and frequency modulation on the right.

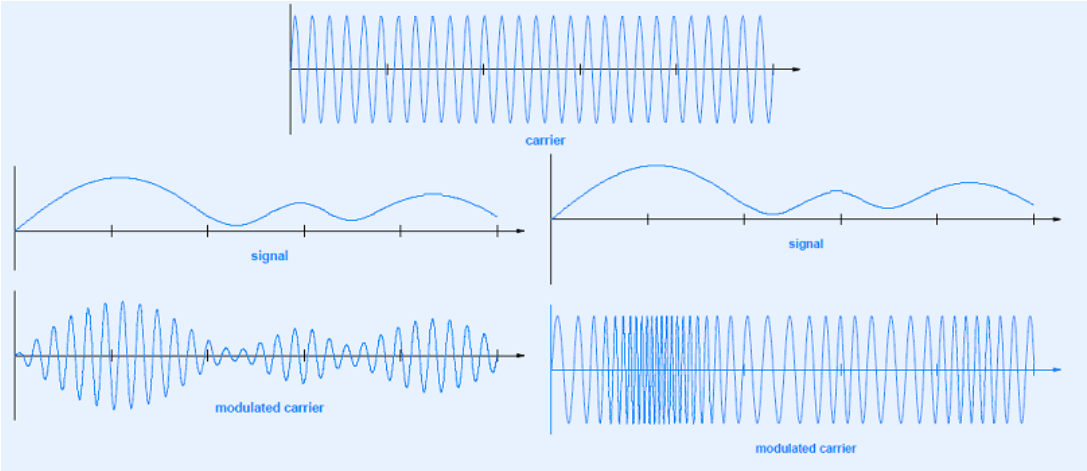

Figure 5. Illustration of Modulation. Source: Comer (2015).

When compared with analog modulation, shift keying techniques allow more discrete values and subsequently more data encoded. A phase shift interrupts the sinusoidal wave to encode data. Data encoding techniques affect the amount of data that is transmitted in a given time. Figure 6 shows amplitude shift keying.

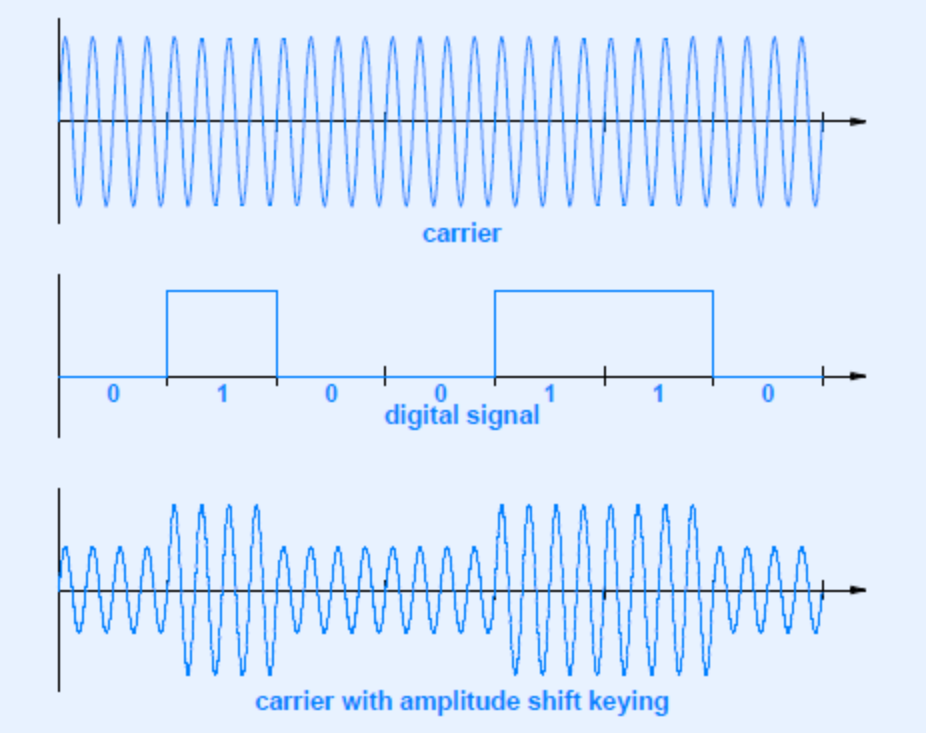

Figure 6. Illustration of Amplitude Shift Keying. Source: Comer (2015).

The importance of data encoding techniques is in their trade-offs with acceptable error rates in a given environment. A signal that uses a more sophisticated data encoding technique will transfer more data than a signal with a lesser sophisticated technique, given that both signals are in the same environment. However, the more sophisticated signal will also be prone to experience more bit errors.

Antenna types and signal strength are physical layer concerns. The type of antenna used influences the EM pattern emitted. Omni-directional antennas commonly emit a pattern that resembles a donut, while directional antennas emit a pattern that looks similar to an uninflated balloon. Power and emission pattern drive signal range in the wireless environment. Signal range is a key variable in creating a connection between any two nodes. Signal range is also a key variable in electromagnetic geolocation. In a military context, an adversary with a receiver will be unable to detect a friendly signal if the signal is indiscernible from the background noise. The transmitter's emission pattern and signal power are therefore major factors in detection.

Physical layer configurations have interactive relationships. Signal strength compared to noise level is directly related to power and antenna type. Bit errors commonly occur due to insufficient signal to noise ratio (SNR) and interference. Physical layer modulation techniques are selected to balance the desired throughput with an acceptable bit error rate in the given environment. However, detecting bit errors is a function of the next layer which is discussed in the following section.

2. Data Link Layer

The data link layer is also referred to as "layer 2" or as the media access control (MAC) layer. Agreements at layer 2 include addressing, maximum frame size supported, and how the medium is shared between users. It is useful to consider again Alice and Bob's conversation at the beginning of this section in order to understand layer 2 functionality.

Addressing includes unicast, multicast, and broadcast. To understand unicast, think of Alice walking into a crowded and loudly saying, "Bob." Everyone not named Bob easily dismisses Alice. Multicast equates to a scenario where Alice says, "Bob, Charlie, and Dick,…" or if she said, "Team 1,…." Anyone not named Bob, Charlie, or Dick would dismiss Alice's call for attention. Likewise, layer 2 broadcasting is similar to Alice saying, "Hello everyone." The addressing function within later 2 is handled by the logical link control (LLC) sublayer.

Maximum transmission size for the network is also understandable using Alice's conversation. Alice frames her sentences using commas, periods, and inflection. Without that framing, her conversation would be difficult to understand. Similarly, data is divided into *frames* at layer 2. The maximum segment size (MSS) dictates how many bytes are transmitted within a frame on the network in question.

The other major agreement addressed at layer 2 is the manner in which many nodes share the same medium.

Imagine if Alice was trying to communicate with Bob from across a crowded room. A common technique would be for Alice and Bob to move closer to one another so

they can hear each other better and will not disrupt—or be disrupted by—other people. The same function needs to occur in data communications. However, two computers often are unable to change their physical proximity to one another. Subsequently, the media access control (MAC) sublayer protocols are designed to avoid, detect, and resolve transmission collisions in wireless networks (Comer, 2015).

There are a few common techniques to allow shared access of the wireless bandwidth that are worth discussing in this section. These techniques can be categorized by the manner in which the medium is distributed, or allocated, between nodes. The shared models include controlled access, random access, and channelized protocols (Comer, 2015). Random access protocols use the idea of competition to determine access to channels. Random access protocols resolve collisions by retransmission (Comer, 2015). Conversely, channelization protocols allocate to nodes a portion of the total resource. The resource is generally frequency and time. Channelization protocols generally need greater awareness of all nodes in order to distribute the resources appropriately. The reservation category allows nodes to share the common resource by employing a central controller to reserve spots, poll for traffic, or without a central controller by token passing. Comer (2015) provides an illustration of the taxonomy of media access protocols in Figure 7. There are a few random access protocols and channelization protocols that prove important to this study and will be discussed further in depth.

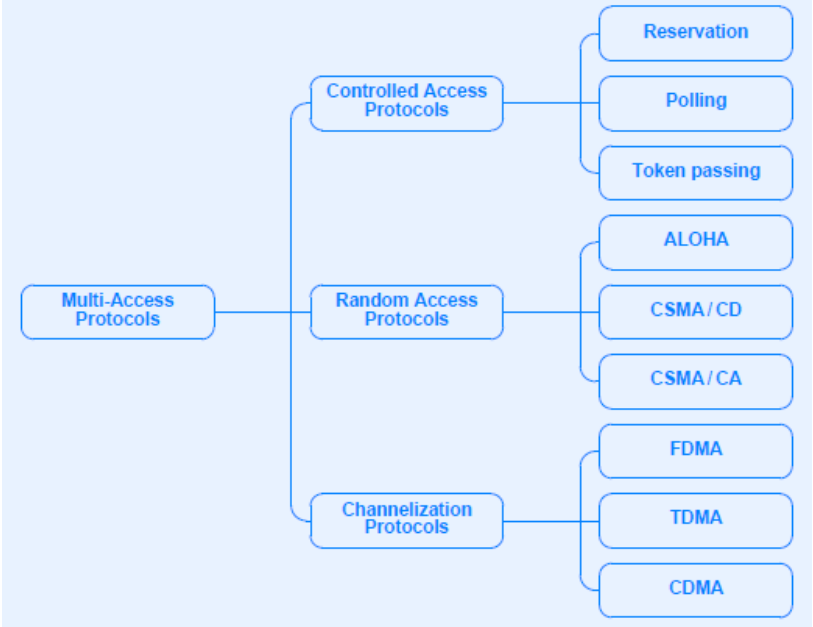

Figure 7. Taxonomy of Media Access Protocols. Source: Comer (2015).

The important channelization protocols include frequency division multiplexing (FDM) and time division multiplexing (TDM). FDM separates nodes by frequency so that communications do not interfere with each other. FDM is illustrated in Figure 8.

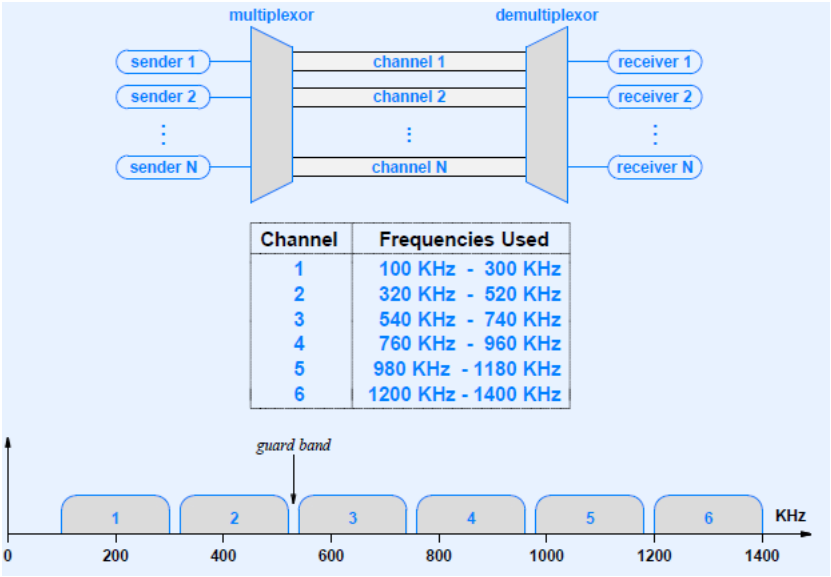

Figure 8. Illustration of Frequency Division Multiplexing. Source: Comer (2015).

Major trade-offs in using FDM is that channels that are unused by the users assigned to them may not be used to provide more bandwidth to users with a greater need than their channel provides. Another technique is to divide the medium by time and give nodes different time slots. This technique is called time division multiplexing (TDM). TDM is illustrated in Figure 9.

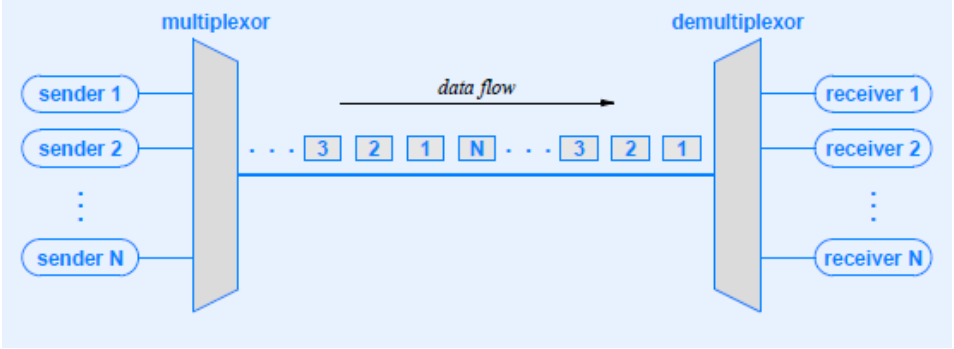

Figure 9. Illustration of Time Division Multiplexing. Source: Comer (2015).

A major trade-off for TDM is the necessary time and effort required to synchronize all nodes and distribute the time slots.

The random access protocols that are important to this study are the Carrier Sensing Multiple Access (CSMA) protocols. The carrier sensing part of CSMA means that the nodes in the network begin by listening for a contending signal so that they do not interfere with ongoing communication (Comer, 2015). This interference avoidance is how the protocol achieves multiple access of the medium. Although all nodes listen first, collisions can occur when two or more nodes need to communicate and they begin to transmit at the same time. CSMA protocols resolve collisions by either Collision Avoidance (CA) or Collision Detection (CD). CSMA-CD is prevalent in wired networks. CSMA-CA is common in mesh networking applications. CSMA-CD performs sub-optimally in wireless applications because of some unique characteristics with wireless signals. There are several common problems in wireless networking that do not exist in wired applications. The first is the hidden terminal problem (Comer, 2015), as illustrated in Figure 10.

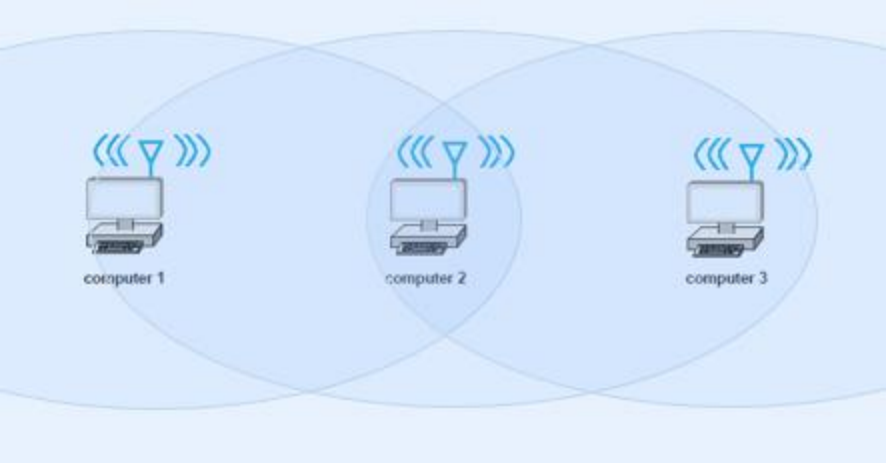

Figure 10. Hidden Station. Adapted from Comer (2015).

As illustrated in Figure 10, computer 1 is in range of computer 2 but will not detect a signal emitted from computer 3. Computer 2 can reach both computers 1 and 3, but computer 3 is out of range to detect a signal from computer 1. With computers 1 and 3 unable to detect signals from each other, both their signals will collide at computer 2. Another nuance in the wireless applications is the exposed station problem. Consider Figure 11. Computer 2 needs to communicate with computer 1 while computer 3 needs to communicate with computer 4. If computer 2 starts first, computer 3 will be unable to begin communicating with computer 4 because it senses that it is exposed to the signal emitted from computer 2.

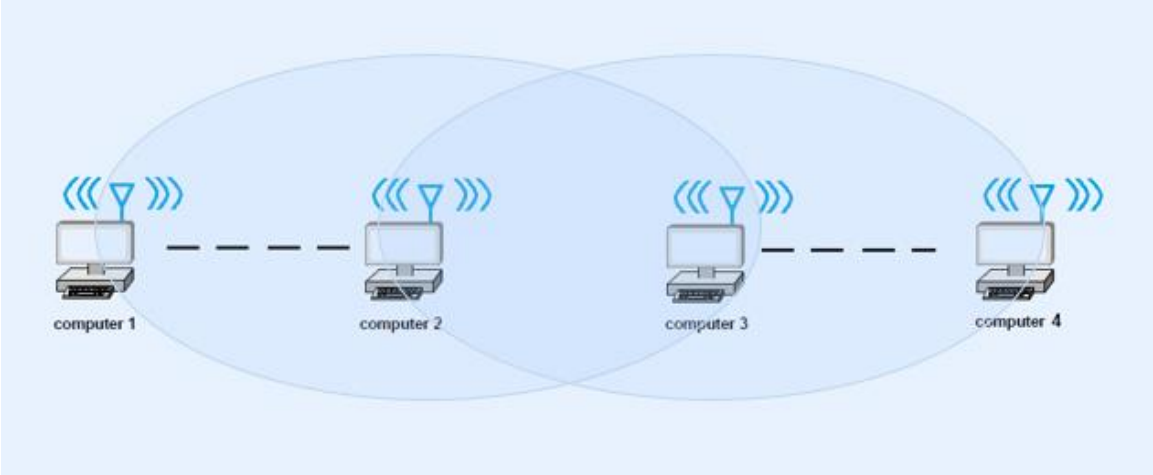

Figure 11. Exposed Station Problem. Adapted from Comer (2015).

The CSMA-CA protocol is designed to help overcome the problems associated with wireless applications by sending ready to send (RTS), clear to send (CTS) and acknowledgement (ACK) messages. CSMA-CA is illustrated in Figure 12.

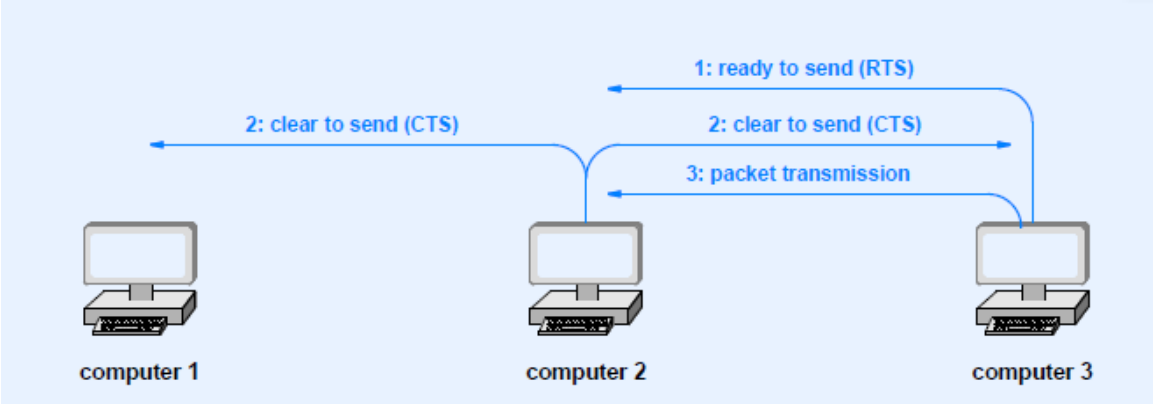

Figure 12. Illustration of CSMA—CD. Source: Comer (2015).

Comer's illustration omits the ACK message of CSMA. In practice, after computer 3 transmits the packet to computer 2, computer 3 would listen for an ACK from computer 2. Receiving no ACK indicates to computer 3 that a collision occurred. Computer 3 would then wait a certain amount of time, and begin again with an RTS message.

The discussed MAC layer protocols are important to this study because they are included in the nodes of the projectiles we prototype and their behavior impacts the type, quantity, and timing of the signals emitted at the physical layer. Layer 2 protocols also impact network behavior in terms of data throughput and latency. For example, if a tactical network requires high throughput, or if there are a lot of nodes within the network, the MAC layer protocols that provide efficient resource-sharing are going to emit a detectable signal while negotiating the resource. Thus, resource negotiations are significant in a tactical scenario because they add to the time that the detectable signal exists which increases the possibility of detection by an adversary. Protocols at even higher layers also add to that time. Layer 3 is discussed in the next section.

3. Network Layer

The next layer in the OSI architecture is called the network interface layer, Internet layer, or simply layer three (Comer, 2015). The network interface layer exists primarily to interconnect networks. To demonstrate using the Alice and Bob example, suppose that Alice knew that Bob was in a group far away, and that she needed to use other groups in between them to deliver a letter to him. Alice needs more than his name. Alice needs an address for Bob and she needs a network capable of transporting the letter to him. The network needs to agree on how to route the letter and then perform routing services. Routing agreements are made at the network interface layer in data communication networks. This section outlines a few relevant network layer protocols in general terms, provides a few examples popular in mesh networks, and then describes their effect on the physical layer.

The most widely recognized implementation of the network layer addressing protocols is the Internet protocol (IP). IP version 4 (IPv4) addresses is detailed in International Engineering Task Force (IETF) publication RFC 791 (1981). According to RFC 791, IP is specifically limited to provide the functions required to deliver a package of bits, called a *datagram*. Mechanisms to assist in data reliability, flow control, and sequencing are found in higher levels. Those required limited functions include addressing and fragmentation, which occurs when the data sent is larger than the

allowable datagram size (IETF, 1981). Refer to RFC 791 for more detail about IPv4 and RFC 2460 for detail of IP version 6 (IPv6).

Protocols that use IP addresses to perform routing are typically designed to achieve maximum throughput. Mesh networks can be broadly categorized by whether they proactively or reactively conduct route discovery and route maintenance (Wang, Xie & Agrawal, 2009). Proactive routing protocols maintain routes for nodes, exchange route and link information between nodes, and have overhead associated with that proactive agreement-making (Wang et al., 2009). That overhead is work that must be done outside of the actual data that users need to send. That overhead also goes through layer 2 down to the physical layer and becomes a detectable signal. Reactive routing protocols discover communication paths only when communication is required by a node. In reactive models, discovered network paths are maintained only during transmission and reception, and quickly expire afterwards. Reactive routing typically requires less overhead when compared to proactive routing models (Wang, Xie & Agrawal, 2009). The trade-off for less overhead often manifests in time delay between sending and arrival at the message's ultimate destination. This delay also means that the physical signal is present and detectable while the route is discovered. Tactically, the more time that the physical signal is detectable, the more likely it is that an adversary with geolocation capability will successfully target friendly forces.

The most popular proactive routing protocol in mesh networking is optimized link state routing (OLSR). Networks using OLSR exchange network topology information proactively by nodes exchanging messages with the state of their links. OLSR-equipped nodes first exchange 'hello' messages in order to discover their neighbors. Nodes then exchange link state messages to share their routes in order to be ready to quickly route data from the application layer. OLSR version 1 is detailed in RFC 3626 and OLSR version 2 is detailed in RFC 7181. We omit detail of OLSR in this section because the proactive exchange of network topology at layer 3 in our proposed disruption-based, bursty, network is unsupportable.

The most popular reactive routing protocol in mesh networking is ad hoc on-demand distance vector (AODV). AODV is detailed in RFC 3561. Networks using

AODV discover the route to link two nodes only after one of the nodes has application-layer data to transmit. The route between the nodes is discovered through a series of route requests (RREQ) and route replies (RREP). Once the route is determined, each node along the route transmits hello messages at short intervals to ensure that the route is active. Any node that does not receive either traffic from the sender to the destination or the frequent hello messages over a given amount of time will transmit a route error (RERR) message. The RERR is then promulgated across the route to let all nodes know that the route is broken. The two nodes that are communicating may then determine that another route is required and would again submit a RREQ. When compared to OLSR, the reactive AODV protocol conducts route discovery when application layer traffic is ready to be sent. OLSR conducts route discovery continuously. In light of the physical signals emitted by a network at layer 3, the reactive protocols for route discovery are preferred for disruption-based networks.

4. Transport Layer

The transport layer is the fourth layer and it resides above the routing layer in the OSI stack. To understand what the transport layer does, it is necessary to consider that a single computer can have many programs running at the same time. While the physical, MAC, and routing layers are designed to get data from the sending computer to the receiving computer, the computers need the transport layer to identify for which program the data is intended. The transport layer uses port assignment to identify the target program. Thinking of the MAC as the recipient's name and the IP as their street address, think of the port as the apartment number at that street address. The port is like the apartment where the program resides.

The transport layer does more than just identify ports. It also indicates how the message should be handled. There are two common methods of message handling. The two common methods deal with whether delivery confirmation is required or not. Think of reliable delivery as a signature service for post mail. When the sender requires signature service, the receiver is asked to confirm receipt. If a package is lost, the sender knows that they must send another. In data communications, most application layer data

must be broken into many small packages. Transport control protocol (TCP) is the name of the reliable delivery service. TCP is designed for the receiver to acknowledge (ACK) receipt. TCP establishes what is called a session. Sessions begin with a synchronize (SYN) request from the sender to the receiver. The receiver acknowledges the SYN request with a SYN ACK. The sender then acknowledges the SYN ACK with an ACK. This SYN, SYN ACK, ACK process is called a three-way handshake. TCP also controls flow so that the data communications do not exceed transportation capacity or the receiver's ability to receive. In data communications this function is called avoiding congestion. TCP does congestion avoidance by manipulating the number of data packages, called segments, sent in between acknowledgements from the receiver. TCP starts with a small number of segments. The small number is called a *window*. The window size adjustment process is called *sliding window*. When the window of segments arrives at the receiver, the receiver sends back an ACK containing the next segment number it expects to be received from the sender. This ACK message is called the *predictive ACK*. The sender then begins to increase the number of segments as long as predictive ACKs are received. When ACKs are not received in a timely fashion, TCP dramatically backs off the number of segments. When ACKs are received again, TCP begins to increase the number of segments again. This description is a generalization. TCP is detailed in RFC 1180 (Socolofsky & Kale, 1991).

The other common transport layer protocol is user datagram protocol (UDP). Unlike TCP, UDP does not ensure reliable delivery and it does not establish a session. UDP is desirable when the sender is not concerned if a package of data (called a datagram in UDP) gets lost along the route. The sender may know that if a few datagrams are lost, more are shortly to follow. UDP messages are common in applications that are streaming data consistently and in messages that are transmitted frequently at each layer of the OSI stack. UDP is ideal for streaming voice and video applications because the receiver will not notice that a frame is lost. UDP avoids the management messages associated with TCP, knowing that users will already be aware of a bad connection and will be likely to adjust the connection on their own. Users may ask each other to repeat what was lost or came in garbled. UDP is also useful in the frequent messages that occur

in order to operate the OSI stack. A few examples of these management messages were mentioned in proactive and reactive routing protocols in mesh and mobile ad hoc networks.

Although TCP and UDP are the most common protocols at layer 4, they are not well-suited for all environments. As mentioned in Chapter II, NASA and other organizations are developing delay and disruption-tolerant networking (DTN). In delay and disruption-prone networking environments, both TCP and UDP are challenged to support the application-layer traffic. TCP, as previously discussed, is based on the idea of a two-way communication session. In disruption-prone environment, an end-to-end connection may not be possible. When the end-to-end connection is possible between sender and receiver, TCP relies upon timely ACKs from the receiver. Without the timely ACK, the sender will keep the number of segments small and will resend them again and again until an ACK is received. TCP, therefore, is sub-optimal for delay and disruption-prone environments. UDP does not fare much better. Like TCP, UDP was designed for end-to-end communication. When the nodes at origin and destination cannot maintain an end-to-end connection, UDP message simply times out at the intermediate routers. In such a case, the sender might continue for some time before realizing that no one is receiving. The DTN protocol is designed as an overlay on the transport layer. Each node along the route uses TCP to store segments, called bundles, from the downstream node and then forwards the bundles to the upstream node when the connection with that node is possible. Like TCP, DTN provides some acknowledgement that the message was received. The DTN protocol goes a step further, adding a custody transfer option between the DTN-enabled nodes along the route (Cerf et al., 2007). With custody transfer enabled, DTN-enabled nodes send custody confirmation messages back to the origin when the bundles are successfully transferred along the route.

5. Presentation and Session Layers

The presentation and session layers in the reference model are not widely used in practice (Comer, 2015). Layers 5 and 6 were added as network management functions for telecommunications providers. Without widespread inclusion, presentation and session

layers do not factor in to this study. They are mentioned here for the sake of completeness.

6. Application Layer

The application layer is offers the most variety of protocols. These protocols are typically specific to the type of application. Application layer protocols are often proprietary and are developed during the programming and development process. The application layer is the layer with which the user—if there is a user—interfaces. Likely the other layers, the application layer directly impacts the time that nodes emit physical signals. The way that a specific application is designed affects the network behavior. For instance, an application that seeks updated information at regular intervals automatically for the user will cause network traffic and physical signal emission. In tactical networks, the users may not be aware that the traffic is occurring. Without the ability to exercise discipline of their emissions, friendly forces are unable to manage their risks.

There is a specific schema offered at the application layer that has impacted mesh networking and command and control efforts. The protocol is called cursor on target (CoT). Developed at the Mitre Corporation, CoT is link-agnostic (Cursor On Target Office, 2013). CoT data is link-agnostic because it can be transmitted between systems that use different network architectures. CoT messages are formatted as basic extensible markup language (XML) language. CoT messages contain the basic data elements "what, when, and where" that can be used for a variety of services (Kristan, Hamalainen, Robbins, & Newell, 2009). CoT data is commonly used to populate common operating system (COP) systems on a map. Important to this study, CoT data can also be used to optimize message routing (Bordetsky and Netzer, 2010). In concept, the COP system map view could also be used to inform the routing function in a mesh network. In lieu of transmitting frequent ink state information across all nodes for manual computation of available links, the same nodes could build a map by using CoT data traversing the network and to predict which routes might be available to their intended recipient. The CoT-enabled nodes would then test the route with messages before transmitting data. Using the application layer CoT messages has the potential to reduce the number of

messages required at the routing layer. The CoT schema therefore fits nicely as a consideration in bursty networking. It offers a possible method to reduce the number of administrative transmissions while determining a given route, which reduces the number of opportunities that an adversary is given to detect and target friendly forces.

B. CONFIDENTIALITY, INTEGRITY, AND AVAILABILITY (CIA) TRIAD

The CIA triad is popular conceptual framework in computer and network security communities. This study uses the CIA triad to inform experiment observations and inform desirable qualities in future disruption-based nodes. Confidentiality, integrity, and availability are well-defined in the Federal Information Security Management Act (FISMA) of 2002. FISMA is legislation that "defines a comprehensive framework to protect government information, operations and assets against natural or man-made threats" (2002). FISMA provides the confidentiality, integrity, and availability as the end goal of information security. FISMA states that information security means "protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction [of information]" (2002). This section first defines confidentiality, integrity, and availability, then describes the interesting relationships between the elements of the triad, and concludes by providing some common network security management practices with the intention of giving unfamiliar readers enough background to understand the study's observations and recommendations.

1. Definitions

NIST special publication (SP) 800–33 defines confidentiality as "the requirement that private or confidential information not be disclosed to unauthorized individuals. Confidentiality protection applies to data in storage, during processing, and while in transit" (Stoneburner, 2001). Confidentiality means keeping information away from those without authority to access it. The goal of confidentiality, as considered in this study, is that an adversary is not able to detect, intercept, and understand the information.

FISMA defines integrity as the "means guarding against improper information modification or destruction, and includes ensuring information nonrepudiation and authenticity" (2002). This is a broad definition that includes many things that are worthy

of discussion. Most generally, integrity means ensuring the information is received without being modified, whether intentionally or unintentionally. Intentional modification by an adversary is extremely dangerous. Imagine a call for fire on a certain grid coordinate. An enemy that can alter those grid coordinates could trigger events with dire outcomes. Unintentional modification is also dangerous. In wireless communications, interference and fading can cause data corruption. In the same call for fire example, undetected unintentional modification can produce the same dire outcomes. FISMA also includes authenticity as a sub-part of integrity. Authenticity means that the information definitely came from the sender. Tailored this study's use, authenticity also means that the sender is a known and trusted part of the friendly force. Interestingly, authenticity also means that the message is not being received a second or third time. This message repetition is called *replay*. In the example call for fire example, replay is also very dangerous. Imagine a replayed call for fire message requesting ordnance on a location that friendly forces have moved into since the original call. Finally, non-repudiation means that the intended receiver is unable to deny having received the information.

Availability is the third element of the CIA triad. FISMA defines availability as "ensuring timely and reliable access to and use of information" (2002). Availability means that forces can access, send, and receive information. An adversary may be able to deny access to information through direct actions like jamming frequency at which data is being transmitted or through indirect actions such as maintaining a significant threat of geolocation and targeting.

2. Relationships

Confidentiality, integrity, and availability have interdependencies that are important. Their relationship is best illustrated through a brief discussion of them as the triad. Although different organizations may prioritize one element over the other two, achieving all three is the information security goal. Over-prioritization can have a negative effect. For example, striving for perfect confidentiality, data and system integrity may naturally limit availability for the users that the network is designed to service. From a threat point of view, factors that impact one goal will likely significantly

impact the other two as well. According to NIST SP 800–33, "[c]onfidentiality is dependent on [i]ntegrity, in that if the integrity of the system is lost, then there is no longer a reasonable expectation that the confidentiality mechanisms are still valid" (2001). Likewise, if confidentiality is lost, there is a high chance that integrity is also jeopardized. When availability is lost, there exists a great risk that confidentiality is lost. If availability was lost through a denial of service or jamming, the adversary likely knows that friendly communications are occurring, which is a ding on confidentiality.

The relationship between confidentiality, integrity, and availability is important in this study. Maintaining confidentiality is a major motivation for designing a network established only through burst. Short-living and highly mobile nodes are naturally more challenging for an adversary to discover than stationary and persistent nodes. Losing confidentiality may mean the adversary will be able to target the force. Ensuring system and information integrity is necessary for successful command and control communications. Commanders and their forces must be able to trust that their information is authentic. They must know that the information is received by their intended recipient and they should communicate in such a way that the recipient cannot deny receiving the information. Achieving availability is also a significant challenge if the nodes are short-living and highly mobile. This study is designed as a proof of concept that availability can be achieved using projectile-based network nodes. The CIA triad thus provides a great framework for evaluating the experiments in this study. The CIA triad as a framework also helps us draw recommendations for future prototypes and desirable qualities in short-living nodes.

3. Security Mechanisms to Achieve the CIA Triad

This section provides a high-level discussion of some of the security mechanisms in place to achieve confidentiality and integrity in persistent networks. Because bursty networking breaks some of the underlying assumptions built into the security mechanisms, understanding both the security mechanisms and their underlying principles is extremely important in this study. The same broken assumptions previously discussed in the OSI-model section are broken assumptions in the security mechanisms put in place to achieve confidentiality, integrity, and availability. Specifically, security mechanisms that rely on a persistent end-to-end are likely to perform sub-optimally in bursty

networks. The next several paragraphs will succinctly discuss symmetric/asymmetric key encryption, and the Diffie-Helman key exchange. Each paragraph provides an overview and some considerations that pertain to this study.

Symmetric/asymmetric key encryption, the Diffie-Helman key exchange, and message authentication codes all use cryptography. Cryptography uses mathematical properties to transform data. It is the basis of encryption and decryption. Cryptography also serves as the means with which data is checked for modifications, detect replays, and authenticate the sender. According to NIST SP 800–177B, cryptography is the underlying mechanism that most technologies use to provide confidentiality, confirm data integrity, and authenticate the data's source, and also to support non-repudiation (Barker, 2016). NIST SP 800–177B provides a more detailed discussion of the following section.

Technologies that employ symmetric key cryptography use the same key to encrypt and decrypt data. In general, symmetric key cryptography means that any two communicating entities must both share the secret key and keep it secret. Symmetric key encryption is faster than asymmetric key encryption (Barker, 2016). However, symmetric key cryptography suffers the well-known key distribution problem. For any two users in a group to communicate confidentially, each user must have share a unique key. The number of keys required grows exponentially as the number of users in the group grows. Providing unique keys to all users is a challenge. In a different implementation, if all users share the same key and the key is compromised, distributing a new key to all users while maintaining the secret is not easy and may take an unacceptable amount of time. Further, shared symmetric key use does not support non-repudiation.

The alternative to symmetric key cryptography is asymmetric key cryptography. Asymmetric cryptography uses a public key and private key for each user. The public key and private key are mathematically related and the mathematical relationship allows the user to decrypt with the private key. The public key is not held as a secret, but is available to all users. Using the Alice and Bob example, Alice can encrypt a message with Bob's public key but the message cannot be decrypted with the public key. The message can only be decrypted with the Bob's private key. Using asymmetric key cryptography reduces the total keys required to two times the number of users in the group.

Asymmetric key cryptography thus reduces the key distribution problem. However, asymmetric key cryptography takes greater computational resources and time (Barker, 2016). There are also hybrid approaches that seek to use the speed of symmetric key cryptography with the reduced number of keys needed in the asymmetric cryptography model. Commonly, asymmetric keys are used to support authentication, achieve integrity, and to generate, agree upon, or transport symmetric keys. These newly established or transported symmetric keys are then used to encrypt and decrypt large amounts of data.

The Diffie-Helman key exchange is a widely used hybrid approach for symmetric key generation between two users (Subramanian, 2010). The goal of generating symmetric keys is to create a fast way of encrypting and decrypting data while ensuring that no one analyzing the data afterwards can compromise confidentiality because the key was not saved by either user and the key never transmitted between them. The Diffie-Helman algorithm uses the same type of mathematical properties as public/private key encryption and decryption. The details of how the Diffie-Helman algorithm works are less important than the underlying assumption that must be true for the exchange to successfully generate a shared secret key. The underlying assumption is the session idea, which relies on a persistent end-to-end connection. In this study of short-living, highly mobile nodes in a network, such a session may not be possible. If an end-to-end connection is possible, the time needed to generate a shared secret key adds to the total time that the nodes in the bursty network must emit a detectable signal. The Diffie-Helman exchange forms the basis for many popular technologies designed to achieve confidentiality, integrity, and availability. These popular technologies include secure socket layer (SSL), top layer security (TLS), Internet Protocol Security (IPSEC), and Internet Key Exchange (IKE) (Frankel, Kent, Lewkowski, Orebaugh, Ritchey, & Sharma, 2005).

This section reviewed the CIA triad as a reference model for the experiment observations and conclusions. The next section discusses the systems theory framework, which informs the design of the experiments as well as the context in which observations are made.

C. SYSTEMS THEORY FRAMEWORK

Systems theory provides an adequate framework for the conduct of the experiments in this study. Systems theory informs the design, the type of data the study collects, and the context in which observations are made. This section defines key systems thinking concepts including contextual knowledge, feedback, and adaptation, which will be central to experimenting with disruption-based networks. It provides a high level overview of several influences that are prompting research in communicating over networks that are fundamentally different. It also examines previous work in the field of networks that are temporary in time and space.

1. The Systems Thinking Lens: From Objects to Relationships

Capra (1996) provides a contextual lens through which to consider the observations and findings of this study. In *The Web of Life*, Capra asserts that all of life is composed of systems. Capra uses the terms *networks* and *systems* interchangeably, as will this study. Capra notes that systems can overlap each other and that networks are found within systems and networks. To demonstrate, Capra shows that organs are systems themselves within organisms, and organisms are in fact systems within ecosystems. Using a wolf as an example, lungs are an aspiration system within the animal, feeding oxygen into the circulation system that feeds the wolf's tissues. The wolf's brain drives the lung system and the circulatory system, while also depending on them. All systems within the wolf depend on each other while also existing because of one another.

A key concept in systems thinking is that properties emerge from the working system or network that cannot be found in any of the system's parts (Capra, p.37, 1996). Applying the same concept to the wolf analogy, the wolf's predatory tendencies and its howl cannot be found in the lung or the brain. Rather, only as a complete system does the wolf exhibit any telltale characteristics. Systems also have a hierarchy, or levels, at work within systems (Capra, p.37, 1996). The wolf does not survive without its pack, which is the next higher level within its ecosystem. Pack behavior emerges from the interdependence and relationships between members, just as ecosystem behavior emerges from the interdependent relations of each of its elements. Capra's point (1996) is that in

any system, pack behavior cannot be found by examining a single wolf. In systems thinking, the pattern of the whole is what must be considered (Capra, p. 37, 1996). Examination of any of the parts will prove to be an inadequate endeavor when attempting to derive overall system properties. Phrased another way by Senge (2006), "dividing an elephant in half does not produce two small elephants." Systems thinking suggests that studying relationships of a system's parts, and studying patterns of a network, provide the value in understanding the behavior of any network.

While easily understood in Capra's "organs within the organism" analogy, the emergent property concept should be understood within a military network context in order to follow the course of this study. In a military context the warfighter, like a wolf, is a system within a system. At a basic level, the warfighter can be viewed as an interdependent set of systems, all relying upon training, to exhibit combat prowess. The warfighter is likewise a part of a hierarchy of superimposed systems: teams, units, commands, and forces. Applied to the military context, Capra's emergent property concept explains that examining the communication networks of the warfighter, the unit, or the command will not fully explain military system's behavior of command and control. As Capra (2010) asserts, network behavior is contextual in nature. Examining any system as a part, separate from its network context, will not explain the system's behavior.

Taking the contextual-behavior concept deeper, Capra asserts that systems thinking is changing the metaphor of knowledge itself (p.39, 1996). He contrasts systems thinking with a historically accepted metaphor of knowledge as a building. In Capra's view, past scientists have used accepted fundamental laws and principles as the foundation of their work. Their work can thus be seen as building up from those foundations. Capra quotes Descartes (p. 38, 1996) to illustrate: "'[the sciences] borrow their principles from philosophy, I considered that nothing solid could be built on such shifting foundations.'" Relating fields of study to architecture is still commonplace, as it is a well-known analogy to first learn the basics, i.e., building blocks, of a given field. According to Capra, systems thinking changes the metaphor of knowledge from one of buildings to one of networks, relationships, and interrelated events. Systems theorists

view reality as an inseparable network of relationships (Capra, p.40, 1996). Capra states that if all natural phenomena are interconnected, one would need to understand all interconnections to explain any single phenomenon (p.41, 1996). Since the task of knowing every variable is daunting, Capra offers an example that a systems approach is still viable through approximate knowledge. To demonstrate that approximate knowledge already in common practice, Capra offers a science teacher dropping an object in front of a class. The teacher provides a formula to calculate the time it will take the object to hit the ground. Capra (p. 41, 1996) offers that although the formula produces an approximate result, it will not be entirely exact. Instead, the class would need to account for air resistance to achieve more accuracy. Additionally, to air resistance depends on ambient temperature and pressure. Capra chases the formula deeper, offering that air pressure also depends on the air's movement in the room. The air in the room can be affected by an open window or even the students' breath. There Capra's analogy stops. Perhaps even before the effects of the observers are accounted for, the value in the approximation is adequate. Ultimately, all interconnections cannot be considered. Capra's point is that learning is still achieved through approximate knowledge even when all interconnections are not considered.

It is central to this study to apply the systems thinking concepts of approximate knowledge and studying an object's interconnections, vice studying the object itself. This research does not attempt to predict the value of employing a projectile in a military network. Instead, the study develops and employs projectile-based nodes in order to observe information flow within a disruption-based network. The projectiles, as objects themselves, and any observations of their qualities are secondary in nature and only exist to inform the primary information flow objective. However, the projectile's inherently ephemeral nature is a key characteristic that allows observations to be made about the effects of its connections in the testbed network. Ultimately, the purpose of this research is to examine information pattern differences in a network physically organized in a fundamentally different way—discretionary in time and space.

2. Barabasi's Party: Nodes, Links, and Clusters

Barabasi, like Capra, is a key influence in systems thinking. Barabasi (2014) provides networking examples that highlight a means with which to perceive the patterns of networks. In *Linked*, Barabasi describes a party with 100 guests, each of whom is invited because they do not know any of the other guests. After a few minutes, Barabasi's guests begin to gather in groups of two or three, exchanging pleasantries and becoming casual acquaintances. In Barabasi's social context, the guests are nodes, their mingling forms links, and groups of interlinked guests become clusters. After a few more minutes, some guests mingle into other groups, thereby connecting several clusters together through the mutual acquaintances of the mingling guest. At this point in the party, Barabasi introduces two pieces of information: that one wine at the party is more desirable than another, and that this information is secret. Sooner than one might assume, the majority of guests at the party know the secret, and Barabasi's preferred wine is depleted.

Nodes, clusters, and links are useful elements with which to examine and describe a networks organization. Clusters are groups of nodes within a network, be they party-goer nodes in Barabasi's example, team member nodes in a tactical unit, or autonomous system nodes acting in a tactical network. A link, then, is comprised dually of the nodes' knowledge of one another and by the nodes' means to communicate with each other. Clusters, nodes, and links can be viewed in a computer network from the different network layers of the OSI stack (Comer, 2014). At the data link layer, computers in a local area network (LAN) are nodes linked through their network interface cards (NIC) to each other through a hub or switch. The LAN is the cluster. Connecting together LAN clusters occurs at the next layer in the hierarchy, the network layer (Comer, 2014). Routers appear as nodes at the networking layer, forming links to other clusters through routing tables and physical media.

In this study, a military force employing a disruption-based network is viewed as a complete system. The data network itself is a system within the military force system. The data network exists as a communication mechanism for the force. The purpose of the force's communication network is to coordinate actions and create shared awareness

among nodes and clusters. Both the data network and the force system are observed in terms of cluster, nodes, and links. The data network's clusters are observed as groups of mesh nodes. Links are the logical connections between the nodes created by the protocols executing at different layers of the OSI stack. The military force system's clusters are comprised of the warfighters working together in teams. Clusters may also include unmanned systems working together as teams. Links are the connections between warfighters or unmanned systems, between teams, and between units within a force. These links are composed of two things: shared knowledge of one another between any two nodes and the shared means of communication between them. Using this framework, and team composed of nodes will have links between the team's nodes and potentially multiple links to other teams.

3. Granovetter's Strong and Weak Ties

Barabasi (2014) uses Granovetter's *The Strength of Weak Ties* (1973) to give a lens through which to perceive how information flows from a given node in a network to a seemingly unrelated distant node. Granovetter (1973) addresses social networks in an attempt to tie micro-level social connections to macro-level social patterns. Granovetter's fundamental question equates to 'how do people's relationships translate into societal trends?' Granovetter analyzes the strength of interpersonal links, defining the strength of a link by a combination of time, emotional intensity, intimacy, and by the mutual services each node provides to each other (1973). Granovetter suggests that when selecting any two nodes from an arbitrary cluster, the strength of the tie between the two nodes is a good predictor of the proportion of interlinks between other nodes. Consider two people in a small social group. If those two people consider each other good friends, Granovetter (1973) suggests that there is a higher likelihood that the other people in the group will all know each other. Conversely, if the two people are merely acquaintances, Granovetter suggests that the other people in the group may not know each other. In a systems thinking sense, Granovetter's hypothesis means that each node in a cluster is likely to be linked to every other node if at least a two of the nodes are strongly linked to each other (p. 1362, 1973).

Barabasi (2014) points out that where Granovetter's hypothesis gets interesting is when the information flow is examined. Granovetter (1973) asked people who changed jobs in a Boston suburb how often they saw the contact that gave them the information that connected them to their new jobs. To Granovetter, the natural line of thinking is that those who share stronger ties will be more motivated to help their job seeker friend find a new job. However, Granovetter's study revealed that only 16% of the job seekers saw their helping-contact frequently. Most of the new jobs (83.4%) came through leads provided by contacts that the job-seeker reported only seeing occasionally or rarely. Barabasi (2014) points out that if all nodes in the cluster know each other, they are also more likely to know the same information. New information, therefore, is more likely to come by way of a weak link from a separate cluster. Although initially it appears counterintuitive that a job-seeker will find a new job through a weak link, by observing information flow it becomes reasonable to think that weak ties can be very important in any network. A biologist might point out that although a bee is not permanent features of a flower beds in a neighborhood, they serve the important role of cross-pollenating. Figure 13 is extracted from page 43 of Barabasi's Small Worlds Link (Chapter 4) in order to illustrate the link strength concept.

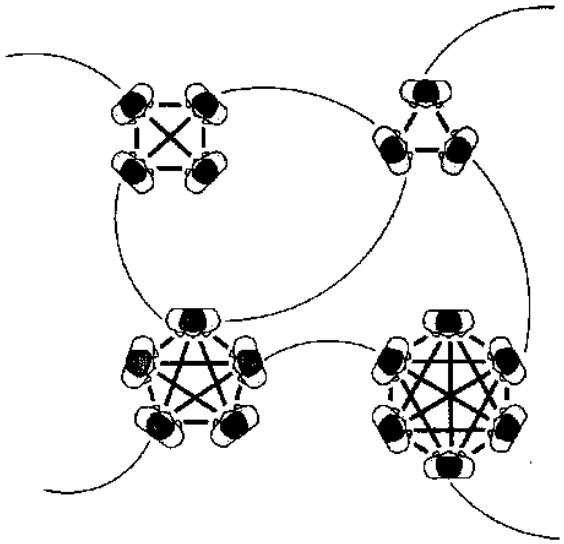

Figure 13. Strong and Weak Ties. Source: Barabasi (2014).

Unlike Granovetter's social ties, the links that form between nodes in an information technology system do not achieve their strength by the length of time that they are connected, or by their emotional intensity or intimacy, or even by the mutual services they provide each other. In this study, weak ties indicate that the link connects separate clusters together. Strong ties are those that can be observed within the clusters themselves. This study uses the strong and weak tie concept as a tool to compare information flow in a disruption-based network testbed with information flow in persistent networks. Through such a comparison, the study evaluates strong and weak tie roles while operating within a disruption-based network construct.

4. Feedback Loops: Weiner's Boat and Viral Videos

A significant element to be observed during this study is that of feedback. In *The Web of Life*, Capra defines feedback in a broad sense as "the conveying of information about the outcome of any process or activity to its source" (1996, p. 57). In systems theory, as Capra points out, feedback exists as feedback loops—either self-balancing or self-reinforcing. Capra (1996) cites cyberneticist Weiner's 1948 boat and steersman as an example of the self-balancing feedback loop. When a boat's path deviates from the steersman's desired course, the steersman pushes the rudder in the direction of the deviation. The boat's deviation from the desired path is decreased, possibly even beyond the optimal point and into another deviation. The steersman reassesses the course, adjusting the rudder to continuously correct the boat's path. Self-balancing feedback loops behave in a goal-seeking manner. Another useful example of a self-balancing feedback loop is provided by Senge in *The Fifth Discipline* (2006). Senge shows that filling a glass of water is a continuous process of monitoring the water level and operating the faucet. As the water nears the desired level, one turns the faucet to slow the water down, and eventually turns the water off at close to the desired level. Figure 14 shows the faucet-filling feedback loop (Senge, p.75, 2006).

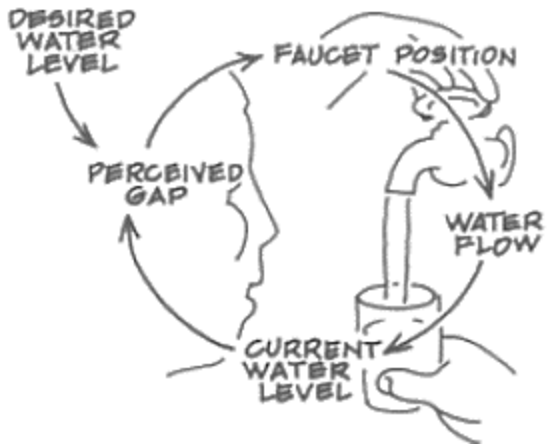

Figure 14. Filling a Faucet. Source: Senge (2006).

Self-reinforcing feedback loops exhibit behavior commonly referred to as vicious cycles (Capra, p.63, 1996). Self-reinforcing feedback loops are also called deviation amplifying and runaway loops. That is, self-reinforcing feedback loops show signs that all causal influences act in the same direction (p.60, 1996). Senge (2006) offers a simple self-reinforcing sales feedback loop, shown in Figure 15. In the causal loop, more satisfied customers lead to more positive word of mouth, which in turn leads to more sales. If an anomaly occurs in the company, say a new product is faulty, and customers are not happy with their purchases, those customers give less positive word of mouth. Sales start to decline because the word is not as positive, which in turn starts to produce fewer satisfied customers.

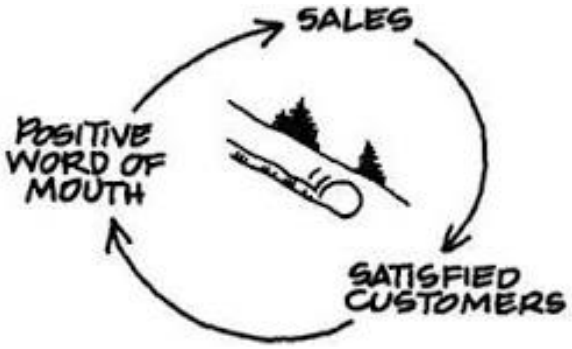

Figure 15. Reinforcing Sales Process. Source: Senge (2006).

A modern example of a self-reinforcing feedback loop is a video or meme that is said to go viral on the Internet. The vast majority of Internet content never goes viral. However, a certain few seem to reach a critical threshold where they begin to become more popular precisely because they are already popular.

Capra credits Maruyama (p.63, 1996) with creating "+" and "-" labels assigned to each causal link. Links whose influence spur actions in the same direction are labeled "+" and links whose influences spur actions in the opposite direction are labeled "-" (Capra, p.60). Labeling is a critical step because feedback loops seldom exist in such simple terms as the steersman, the faucet, and the sales process. A given system may have many causal links in a loop, and many loops may exist within the system. As Capra states, feedback loops are "abstract patterns of relationships embedded in the activities of [systems]" (p.64, 1996). As a simple rule, Capra states that a feedback loop "will be self-balancing (-) if it contains and odd number of negative links and self-reinforcing if it contains and even number of negative links" (p. 61, 1996). This study will search for critical feedback loops and attempt to trace causal links in the disruption-based network testbed in order to examine and explain network behavior and information flow.

The feedback loops observable in this study exist within the layers in the OSI architecture (Comer, 2011) and also exist within the cognitive domain. A feedback loop exists at the physical and network layers, where the EM signal of a given mesh node (N_1) is received by another mesh node (N_2). N_2 then decides what to do, choosing either to ignore, respond, trust or challenge for authentication, repeat or route the received signal. Each of these actions depends on the protocol of the mesh node, and the action itself is the next step in the feedback loop. N_2's response signals back to N_1. The network clusters converge over the link during flight, transmit critical information, and then effectively break back into separate clusters once the projectile is destroyed. In chapter IV, this study proposes a model in which perceived gaps in critical shared knowledge act as the feedback for creating the network via a burst. These bursts then allow for information to flow, which closes the perceived gap in critical shared knowledge.

5. Adaptation

Feedback is closely related to the systems property of adaptation that this study will examine in depth. Adaptation, in fact, is a function of feedback (Capra, p. 56, 1996). Capra writes that the interplay of feedback and adaptation is that the first link is affected by the last in the feedback loop. Feedback is what drives system adaptation.

Creating the data network, in the context of this study, is a deliberate choice and an adaptation. The projectile operator has the responsibility to choose the moment to create a larger network by connecting distributed clusters. The network clusters converge over the link during flight, transmit critical information, and then effectively break back into separate clusters once the projectile is destroyed. The burst transmissions allow for command and control communications. Those communications allow for the force to conduct decision-making and to ultimately adapt to the tactical scenario.

6. Delay

Feedback loops rarely exist without delay. Sterman (2010) states that delays are processes whose outputs lag behind their inputs. Delays commonly have dramatic effects in the results of feedback loops. Sterman (2010) describes material and information delays, modeling first-order and higher-order examples of both. Delays tend to make elements in a feedback loop overshoot the optimal solution, which produces oscillation (Sterman, 2010). Weiner's (1948) steersman example, given as one of the first in systems thinking and remaining one of the most simple, includes a delay. Capra (1996) mentions that the steersman may actually push the rudder long enough to go through the optimal course correction, and only after the steersman perceives the new deviation is another countersteer possible. Senge (2010) provides a useful example of the oscillation produced by delays. Consider person beginning to shower. The person makes an initial guess at the right temperature setting by turning the knobs on. Cold water initially shoots from the shower head. Hot water is delayed by the cold water that remains in the pipes. Impatient and uncomfortable, the person in the shower turns the knobs hotter, only to experience scalding-hot water in a few moments. Now perturbed by the hot water, the shower

reverses the knobs to a colder setting again. Figure 16 shows Senge's self-balancing feedback loop with the added delay element.

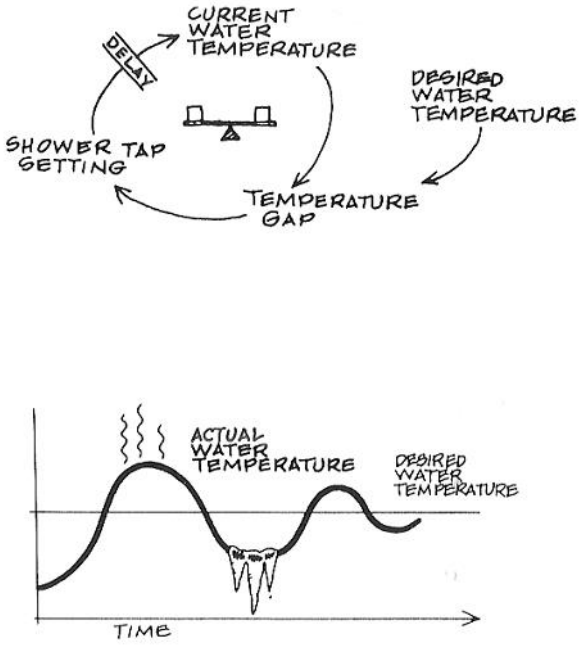

Figure 16. Balancing Process with a Delay: A Sluggish Shower. Source: Senge (2006).

Our initial intuitions are that delays are an important variable in bursty tactical networks. Perhaps networks with persistent connections became popular because they minimize delay. However, persistent connections are not feasible in the future operating environment. Therefore, this study will observe the effect of delay in the information flow captured during experimentation.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. EXPERIMENT DESIGN

This chapter describes the study's multi-part discovery experiments and the methodology used to design them. We designed and conducted experiments that were sufficient to explore the creation of an actual physical network just in time to accommodate the bursts of application layer data required for command and control.

We planned our discovery experiments in four phases. During Phase I we planned to conduct a feasibility analysis. In Phase I, we hypothesized that we could create a suitable prototype by carefully selecting of a very small mesh radio, power supply, and microprocessor and inserting it as a payload into 3D printed assembly of our own design. Phase II consisted of discovery experiments designed to test the hypothesis that simulated command and control information would be retrieved from a remote node positioned out of communications range using the short-burst of the projectile during flight. We wanted to test whether we could create a networking burst that would exist for just long enough for the decision maker to receive the message from the remote node. Phase III consisted of more advanced discovery experiments. We designed Phase III to test the hypothesis that we could use a burst in the network to transmit movement instructions to a UGV and visually confirm the UGV's movement. The UGV's movement would prove that actions can be executed between the bursts, effectively beginning a feedback loop of command and control. The Phases II and III hypotheses were designed to prove the concept that it is possible to physically create and use a network during short bursts. We dub this approach *distrupted tactical networking*. Our task required us to condense all layered traffic in order to support requirements driven by the command and control processes during a given tactical scenario. Phase IV consisted of demonstration experiments, illustrating a few of the possible implementations of disruption-based networks in the future operating environment. The demonstrations were planned to be simple animated vignettes designed to give the reader a better picture of the networking by burst idea. We refined our prototype through all four Phases of our campaign of experiments. Figure 17 shows our experiment campaign design.

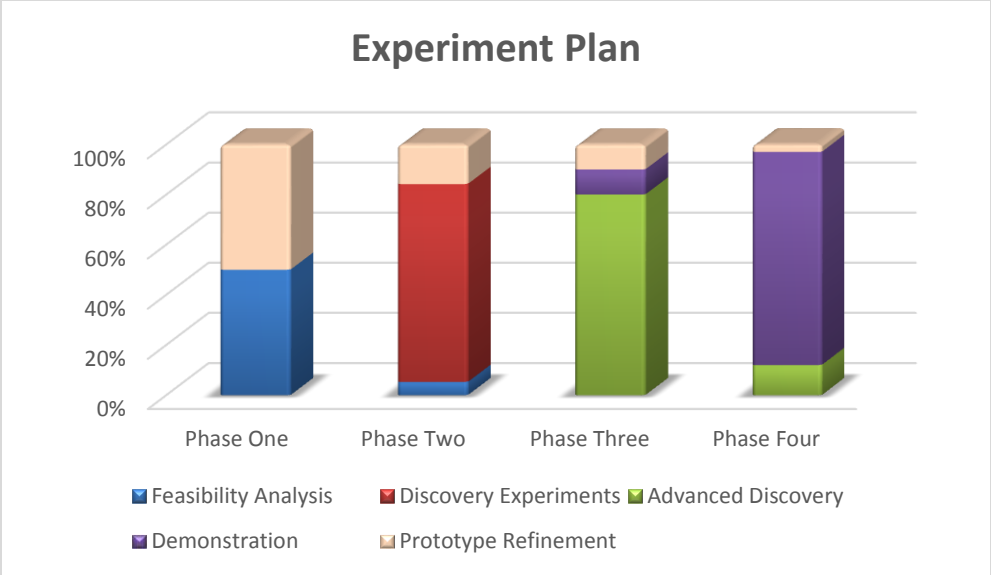

Figure 17. Experiment Campaign Design

We designed the campaign of experiments using the multi-space criteria model. Section A details our experiment design process while section B discusses the final design of our experiments.

A. MULTI-SPACE CRITERIA MODEL

The multi-space criteria model includes design space, functional, and criteria space constraints (Figure 18). These constraints define the feasible solution set for creating a suitable prototype for additional experiments (Statnikov & Statnikov, 2011). The design variables are the independent variables under the immediate control of the systems designer (Alberts, 2002). Design variables impact the criteria achieved. Functional constraints also impact the criteria achieved by the experiment. Functional constraints are those that must be accepted by the experiment designer. The experiment designer does not control functional constraints. Functional constraints may be outside the scope of the study, appear randomly, or may be necessary to accept in order to conduct the experiment. Together, design space variables and functional constraints produce criteria variables. Criteria are the observed results of an experiment. Criteria space constraints are the results that will provide sufficient measure for the experiment's purpose. Figure 18 depicts a geometrical interpretation of the PSI method.

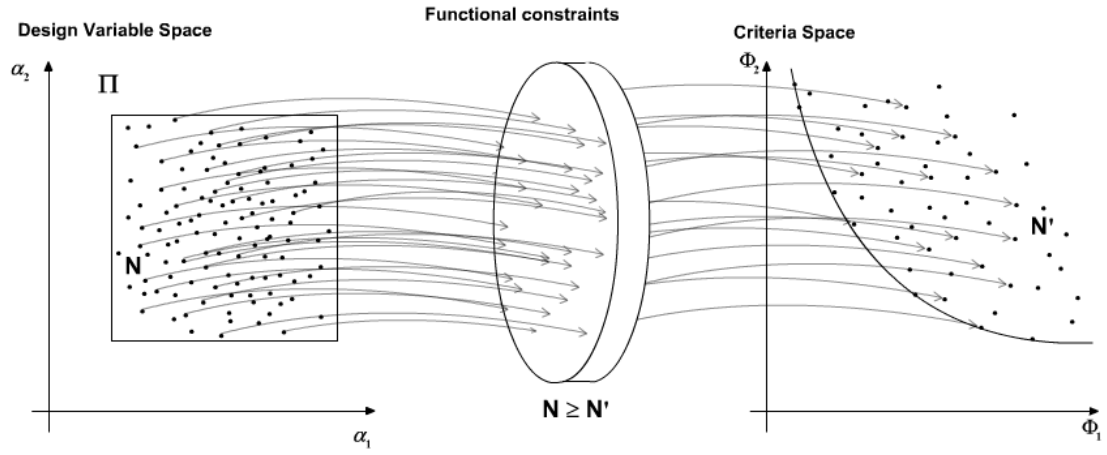

Figure 18. The Geometrical Interpretation of the PSI Method. Adapted from Statnikov & Statnikov, 2011.

In Figure 18, α_1 represents the first design space constraint. α_2 represents the second design space constraint. Together, α_1 and α_2 form the feasible area of the design space containing several solutions (N). N solutions are then functionally constrained, which further reduces the number of suitable solutions. Finally, fewer solutions fall within the feasible area of the criteria space constraints. In Figure 18, Φ_1 and Φ_2 represent single constraints. In most experiments, this one notwithstanding, more than two constraints exist in the design space and the criteria space. Figure 18 illustrates the concept in a way that will help frame the constraints this study considers while creating the projectile-based prototypes and designing the discovery experiments.

Table 1 shows the design space, functional, and criteria space constraints in this study. The following sections detail how each constraint impacted the study.

Table 1. Design, Functional, and Criteria Space Constraints

DESIGN SPACE CONSTRAINTS	FUNCTIONAL CONSTRAINTS	CRITERIA SPACE CONSTRAINTS
Mash Radio Type	Time Available for Study	Data Transferred
Microprocessor	Testing Sites	Time Connected
Launcher Type	Number of Launchers	Flight Range
Descent-Control System	Commercially Available Nodes	Signal Range
Network Topology	Component Features (protocols, power, antenna type)	Time of Flight
	Weather Conditions	

The following portions of this section detail how each constraint impacted our experiment design and ultimately shaped the results we obtained.

1. Design Variable Constraints

The design space constraints we considered while creating the projectile-based mesh node prototype include the components that control the node's behavior. These variables include: a) the mesh radio, b) the microprocessor, c) the descent control mechanism, d) the launcher type, and e) the network topology.

a. Mesh Radio

We selected Virtual Extension (VE) 209S mesh radios for use in our experimental networks (Virtual Extension, n.d.). VE mesh (VEmesh) radio modules measured 38.1mm by 21.6mm and were light weight. VE mesh modules connect to nodes through UART connections and the RS-232 protocol. The VE mesh modules ran on 3.6 volts direct current. Our VE gateway connected through USB to our Windows-based work station. Virtual Extension provides a programmable interface that enables the network manager to set variables such as baud rate and number of hops. Virtual Extension also provides a graphic user interface (GUI) for data collection through the gateway. Figure 19 shows the VE mesh gateway and mesh radio modules.

Figure 19. Virtual Extension Components. Source: Virtual Extension. (n.d.).

The VE mesh network uses TDMA at layer 2 to do frequency hopping spread spectrum (FHSS) in the sub-GigaHertz industrial, scientific, medical (ISM) band. VE mesh's novel approach to routing at layer 3 is to use Simulcast. Figure 20 shows the Simulcast concept.

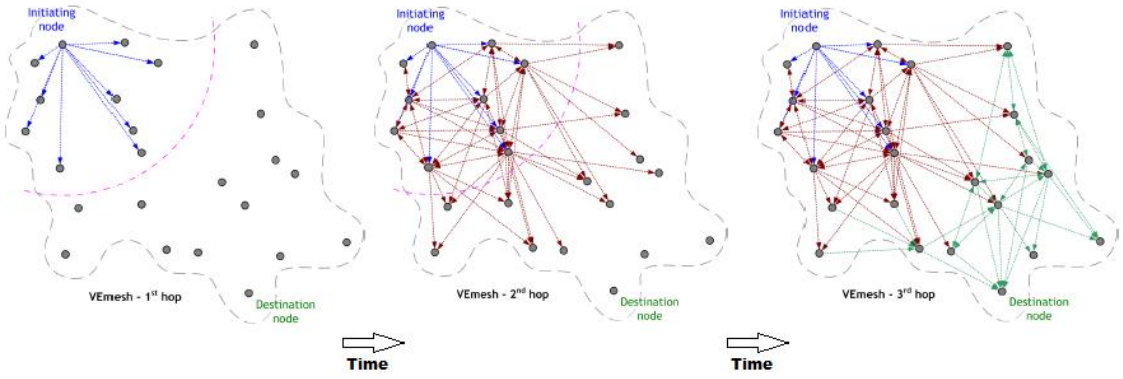

Figure 20. Simulcast. Source: Virtual Extension (n.d.).

Simulcast eliminates the need for routing table maintenance and the overhead associated with determining a single propagation path through the network. Instead of routing tables, each node synchronizes onto a pseudo-random sequence of hops inside the total bandwidth. Figure 21 illustrates analog bandwidth. Consider each bar to represent one of the possible hops by the mesh radios.

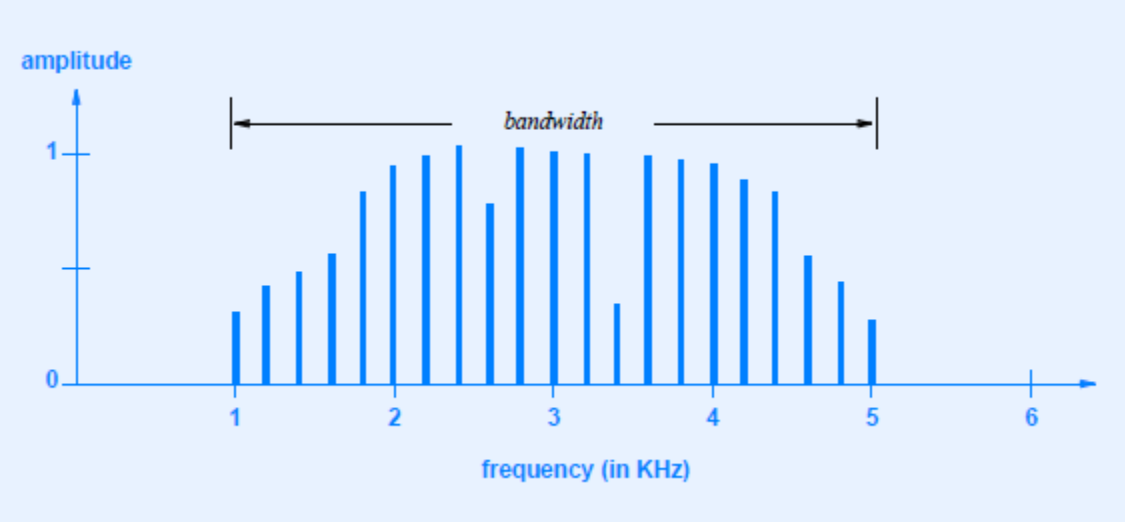

Figure 21. Definition of Analog Bandwidth. Source: Comer (2010).

When no nodes are transmitting, all nodes listen in synchronization for a few milli-seconds (ms) at the frequency in the sequence. When the initiating node transmits, all other nodes listen and nodes within range of the initiating node receive the datagram. Those nodes then retransmit in the next time slot at the next frequency—simultaneously. The initiating node listens for the retransmission to confirm it was received, and stops sending that datagram. In Figure 20, the VEmesh—3rd hop shows the subsequent nodes hearing the retransmission, and then retransmitting again in the next time slot at the next frequency.

As design variables, the VEmesh module's characteristics made it capable of providing our criterion variables—the experiment's results. Specifically, their low cost and small size allowed us to place them inside three-dimensional (3D) printed projectiles. Their low power consumption allowed us to select a small, 3.6 volt lithium ion battery to

include in the projectile. For example, some of the VEmesh modules' other characteristics also acted as constraints in our experiments, markedly impacting our results. The RS-232 protocol allows for a small number of Bytes per datagram, which limited the amount of data we could successfully transmit during the brief flight. Additionally, the VEmesh network's TDMA and FHSS protocols required significant time for synchronization in experiments in which we powered on nodes for the first time during the projectile shot. The synchronization time was controllable by selecting the number of frequencies to hop between. The default was 20 hops which measured 10–14 seconds for synchronization during our experiments. We set the hops to zero, but were unable to reduce synchronization time to below eight seconds, which was the effective time of flight. Through personal communications with VEmesh, we discovered that the 5 volts is eased onto the circuit upon start up. Rolling the power out the the circuit slowly protects the VEmesh modules' components, but negatively impacted our ability to communicate while the projectile was in flight. Thus, both synchronization time and power-on cycle limited the total time available to transfer data.

The VEmesh gateway we used also had a unique effect on the criteria space. Our VEmesh gateway connected via a universal serial bus (USB). We attempted to monitor network behavior using WireShark to monitor the USB port. Our results were interesting. Although we could determine the time and type of network commands, the PCAP record indicated that network traffic occurred every few milliseconds. Those milliseconds were due to the USB port interfacing with the Windows computer, rather than the radio conducting network functionality. Our PCAP file also failed to capture the layer 2 TDMA synchronization traffic. VEmesh gateway models that connect via Ethernet would provide much better observation of network traffic.

b. The Microprocessor

The next design variable we selected was the microprocessor to couple with the VEmesh modules. We selected Arduino™ Pro Mini shown in Figure 22.

Figure 22. Arduino Pro Mini. Source: Arduino (n.d.)

Arduinos are small, cost-friendly, easily programmable, open-source platforms that provide a lot of flexibility in use (Arduino, n.d.). The Arduino is not a computer with an operating system (OS) capable of running applications and has finite internal memory. Arduino provides a programming interface as well. Programs written in the Arduino interface are called sketches. In our first experiments, we designed a sketch that included an accelerometer, barometer, and tilt-sensor. We designed our sketch to collect flight data from the sensors on ascent, sense apogee, and then act as a relay on descent. By sensing apogee, we instructed the Arduino Pro Mini to deploy a parachute. As a design variable, the Arduino also impacted our criteria. Our initial design was to transmit flight data through VEmesh to the gateway as the Arduino's sensors collected it. This limited the duration of flight time that would be used to transmit data to or from remote nodes. If the Arduino was an operating system, flight characteristics would be easily stored locally for retrieval after descent.

c. The Descent Control Mechanism

The descent control mechanism was a design constraint in this study. We chose a parachute. The first prototypes had a 24 inch nylon parachute. Later models had a 48 inch parachute to accommodate greater size and weight. A descent control mechanism is not a required feature in a projectile-based network. Hypothetically, projectiles could be destroyed after each use or left to operate from the ground. However, destroying the prototypes in this study would limit the number of observations possible. On average, each projectile took more than 10 hours to 3D print and assemble. Although we included a descent mechanism for our nodes, disposability may prove to be a desirable quality in

future projectile models. It seems equally likely, though, that the parachute could be a desirable feature to extend service time for data communications.

d. The Launcher

The projectile's launcher was also a design variable, as well as the angle of deployment and trajectory in relation to the remote node. We envisioned using a standard M203 grenade launcher. However, concerns about access, safety restrictions, and firing range time led us to search for suitable alternatives. We first examined the range we would need for our projectile. The desirable flight range was informed by the range of our signals. Our VEmesh node signal range, given the single wire antenna and 3.6V battery, was less than 200 meters. We looked for launchers with ranges from 100 to 400 meters. We built a standard spud gun from 2" and 4" PVC pipe but we experienced an unacceptable power variance with our initial prototypes. Subsequently, we found the pneumatic line thrower (PLT) by Restech Norway (Restech Norway, n.d.). The PLT family of products is designed for sea-based operations including vessel to vessel, anchoring, man overboard, and whale tagging. The Restech Rescue 230 has an advertised 230 meter minimum range. The PLT Mini has a range of around 100 meters. We decided to use both the Rescue 230 and the PLT Mini. Figure 23 shows the Rescue 230 and Figure 24 shows the PLT Mini.

Figure 23. Rescue 230. Source: Restech Norway (n.d.).

Figure 24. PLT Mini. Source: Restech Norway (n.d.).

e. Network Topology

Network topology was an additional design variable in Phase II and Three of our experiments. Specifically, we chose placement of the VE Mesh gateway and base station, the number of clusters and nodes. In Phase II, we determined that placing a single remote node out of signal range from the base station was sufficient. We programmed the Arduino Pro Mini in the remote node with simulated critical command and control information. The projectile in our first experiment would pass through a point where it would maintain a signal with the gateway and make a connection with the remote node. Figure 25 is a visualization of the experiment concept.

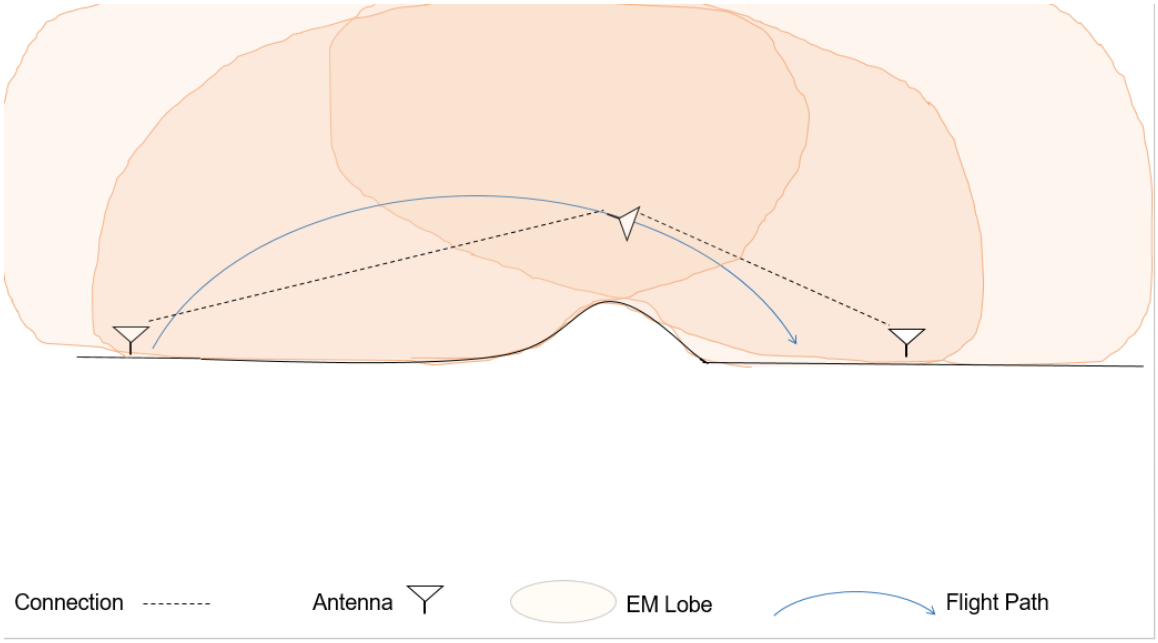

Figure 25. Connection Time Visualization

According to Alberts (2002), the tactical scenario can be adjusted to provide the opportunity to observe the criteria space variables systematically while manipulating the design variables. We planned to adjust our tactical scenarios to observe the criteria while manipulating the design variables. Those adjustments are captured in detail in the next chapter.

2. Functional Constraints

The functional constraints are variables that are accepted by the users of the system or environmental factors (Alberts, 2002). Alberts (2002) dubs functional constraints as intervening variables. Functional constraints impact the relationship between the design space and criteria space variables. This study was impacted by several functional constraints.

We accepted that the scale of our experiments had to remain small because of several general constraints. At the most basic level, this study had to be completed during the course of 18 months. If additional manipulation of design variables is necessary, we recommend that it be done in future work. We also had limited access to sites where we

could fire the pneumatic line thrower. We also lacked the ability to scale up any scenario because of the number of pneumatic line throwers available. While we could form clusters of terrestrial-based nodes, interconnecting them with the use of a projectile was naturally limited. We therefore chose to use a two-cluster experimental design.

Another functional constraint we accepted was the lack of a commercially available bursty-networking node prototype for our use during the study. We attempted to test Rafael's FireFly and AeroVironment's BlackWing, but were unsuccessful in gaining access for testing purposes. We therefore planned Phase I as a feasibility analysis test whether we could prototype our own. We 3D modeled several body assemblies, printed the assemblies using Ultimaker 3D printers, and used commercially available components to make a working prototype. We also accepted the mesh networking protocols, routing protocols, and security protocols in the mesh radios that we selected. Changing these protocols to observe their effect on the criteria space variables was not feasible because of work capacity and time. Similarly, we accepted the antenna type and power at the receiver and transmitter as functional constraints. Those variables play an integral role in the signal range but are already contained within the mesh radios available for use in this study.

3. Criteria Space Constraints

Criteria space variables are the dependent variables that are the outputs. According to Alberts (2002), criteria space variables are the products of the system, representing the behaviors that are important to the success of military operations. A criteria space constraint is the output that is required to answer the research questions and meet the objective of the study. As a proof of concept thesis, our primary criteria space constraint was that we needed to confirm that command and control information could be successfully communicated through the projectile. Command and control information is application layer (layer 7) data. We predetermined several criteria space variables that would help inform us about desirable qualities of a bursty networking node to help us observe the flow of information in a disruption-based network.

Our criteria space variables during the experiments included flight range, signal range, time of flight, and the time of connection. The time of connection in this study is defined as the duration of time when the receiving nodes receive the transmitting node's propagated signal and the SNR is sufficient to permit data transfer (Stanley and Jeffords, 2006). Time of connection can be visualized during flight as the time at which the receiving node enters within the lobe of the transmitting node's propagated signal through the time that the receiving node departs the transmitting node's lobe or the time at which the projectile touches down on the surface. Figure 25 illustrates the connection time variable. Time variables are observed in seconds. Data received is also a criteria space variable in the discovery experiments. Collected in bytes, data is observed at the base station co-located with the deployment mechanism and the shooter.

In subsequent experiments, time of connection and data transmitted remained as observable variables. However, the criteria space constraints expanded. The study sought to transmit application-level data over the disruption-based network. Additionally, criteria space constraints included observations of the level of shared understanding by the nodes and clusters. While observable with transmitted data stored in unmanned systems and subsequent node behavior, observing shared understanding among human tactical role players and teams would require prior tasking to make them record their understanding during the experiments and post-experiment collection during after-action sessions.

4. Relationships Between Variables

The study expected to reveal several relationships among the variables. Beginning with the criteria space variables, data transmitted is proportional to connection time. Likewise, time of flight can be proportional to connection time, given that the vector of flight in relation to the separated nodes and clusters supports the relationship. That is, if the flight trajectory brings the projectile within range of the separated node, additional flight time within signal range will allow for a greater amount of data to be transferred. However, if the trajectory takes the projectile in a direction away from the separated node, greater flight time will not influence the amount of data transferred. We also

expected to see a direct relationship between protocols inherent in our components and the overall network convergence time. The next section provides a detailed description of the phases in our campaign, beginning with an overview and finishing with a subsection for each phase.

B. PHASES OF EXPERIMENTATION

We began Phase I by selecting components to include in our feasibility analysis. We then created projectile-based mesh nodes that are suitable for testing in subsequent discovery experiments. Next, we tested the prototypes for suitability and flight characteristics. Once we determined that we had a functioning prototype, we proceeded to Phase II and conducted simple discovery experiments. In the first experiment, we co-located the node, the gun, and the gateway. We shot the prototype toward a single remote node positioned out of range but attempting to transmit data back to the gateway. The projectile acted as a single hop, connecting the remote node and the gateway. Phase III included in an event where we were able to transmit command and control instructions to an autonomous unmanned ground vehicle (UGV). Phase III served primarily as advanced discovery experiments, where we confirmed our observations from Phase II and gained more detailed observations about information flow in a short-living network. The simple experiments during Phases II and III are a simple proof of concept.

We concluded our experiment campaign with Phase IV. During Phase IV, we created several demonstration vignettes to illustrate the overall potential of the disruption-based networking concept and some of the possible applications. Subsequent paragraphs detail the prototyping process, the initial discovery experiments during Phase II, advanced discovery experiments during Phase III, and detail the vignettes created during Phase IV.

1. Phase I—Feasibility Analysis

Phase I was a feasibility analysis. Without commercially available assets to use in our networking-by-burst experiments, we hypothesized that it was feasible to create our own using readily available, inexpensive components. Our task was to produce a working network node capable of achieving the criteria during our experiments. In order to

produce such a node, we needed to combine the design constraint variables of our multi-space criteria. Phase I design constraint variables included the mesh radio, micro-processor, descent control mechanism, and launcher. Since each selected design variable would inevitably introduce functional constraints, we were especially cognizant of our criteria constraints and sought ways to simplify variables in order to produce just what was needed in order for us to observe sufficient criteria. The prototyping process was broken down into two concurrent lines of effort. Those lines of effort were 1) component integration and programming, and 2) assembly design and creation. Both efforts evolved incrementally. Decisions we made in one line of effort led us to adjust the design of the other effort. Challenges we discovered in assembly design impacted our component integration effort. Likewise, issues we discovered with integration and program drove the evolution of our assembly design. The following paragraphs are a narrative of the prototyping process and our feasibility analysis.

We began with a very basic design for Prototype 1. We built a common combustion-type spud gun with a 50.8 (2 inch) inner diameter. Prototype 1, depicted in Figure 26, included a payload bay for the electronic components and micro-servo. The micro-servo held a rubber band that kept the parachute in the parachute bay by holding the base cap. Our programming line of effort began with a simple timer sketch in Arduino's IDE. The sketch is included for reference purposes in Appendix A. The timer sketch provided a time delay after power up that allowed us to test Prototype 1's descent mechanism after launch. We used Trimble's open source 3D modeling software, SketchUp, to quickly produce standard tessellation (.stl) files for 3D printing. We printed Prototype 1 using an Ultimaker 2+ 3D printer. After assembly, we proceeded to bench tests.

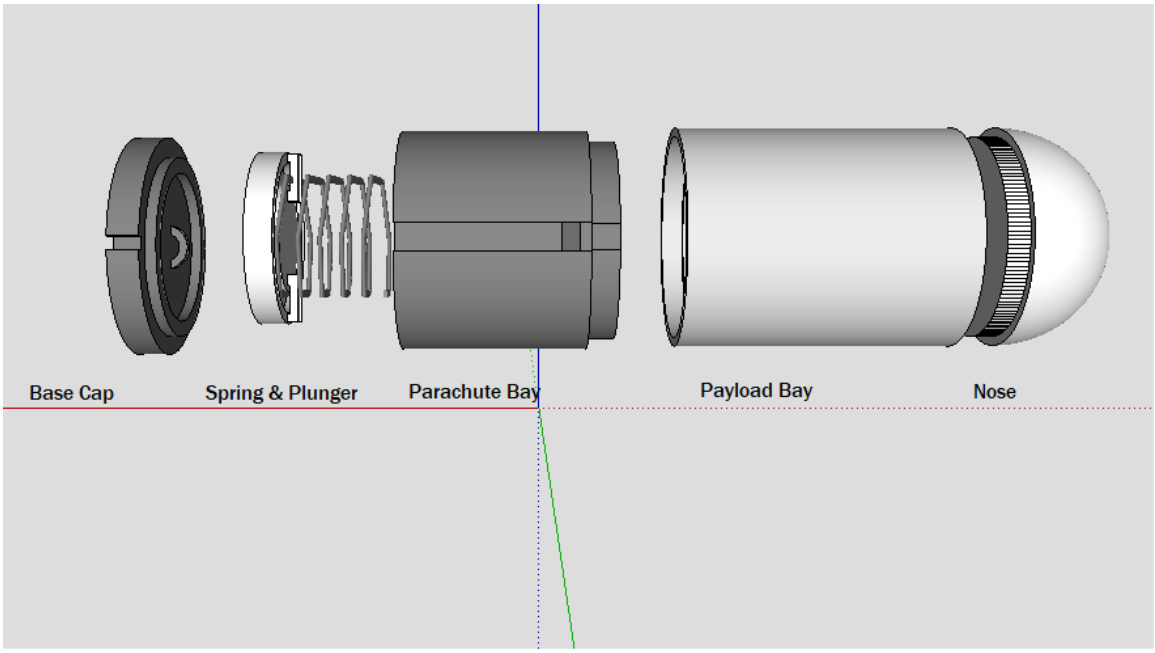

Figure 26. Prototype 1: Developed in SketchUp

Bench tests with Prototype 1 showed us that the rubber band was not strong enough to compress our spring. The rubber band also occasionally caught in the relief channels and the barrel. Concurrent launch testing also demonstrated that the spud gun we created did not produce consistent power to shoot the projectile along a reproducible path. Power differences were noticeable between shots. We were able to discover the power reliability problem in the spud gun even before installing the payload into the assembly. We began designing Prototype 2 with the tasks of selecting a new suitable launcher and subsequently redesigning the descent mechanism and projectile assembly.

Prototype 2 was the first model designed for Restech Norway's pneumatic line thrower (PLT). PLT's barrel reduced the possible outer diameter of our projectile from 50.8mm to 38.5mm. Because we did not believe that the parachute would condense inside the 38.5mm diameter, we envisioned a parachute door on the side of the projectile. We designed the upper portion of Prototype 2 to remain 50.8mm outer diameter. The door was held closed through direct contact with the micro-servo.

While testing Prototype 2's flight characteristics of, we discovered that the polylactic acid (PLA) housing failed to remain intact when shot from the PLT, at least at

the in-fill density of our prints. PLA is a biodegradable plastic that is quite common in additive manufacturing. PLA has a low melting temperature and resists shrinking and warping better than some of the other 3D printable plastics. We printed Prototype 2 with our Ultimaker 2+'s default settings for normal print quality (22% in-fill). Prototype 2 is depicted in Figure 27.

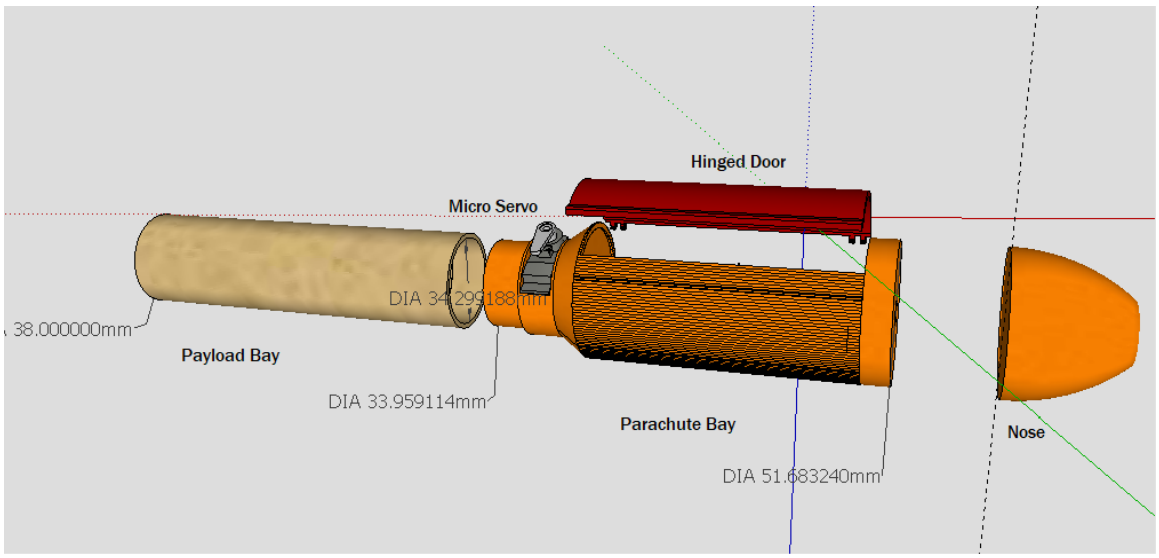

Figure 27. Prototype 2: Developed in SketchUp

We adjusted Prototype 3 by returning the parachute bay to the aft of the projectile, accepting the reduced 38.5mm maximum usable diameter. In lieu of the rubber band system we added two rods on the plunger assembly to be held directly by the micro servo. We printed three Prototype 3s. We printed the first at 22% in-fill and the last at 100% in-fill. We were confident of the flight characteristics with Prototype 3 but wanted to verify that the PLA would remain intact against our compressed air charge. Prototype 3 is depicted in Figure 28.

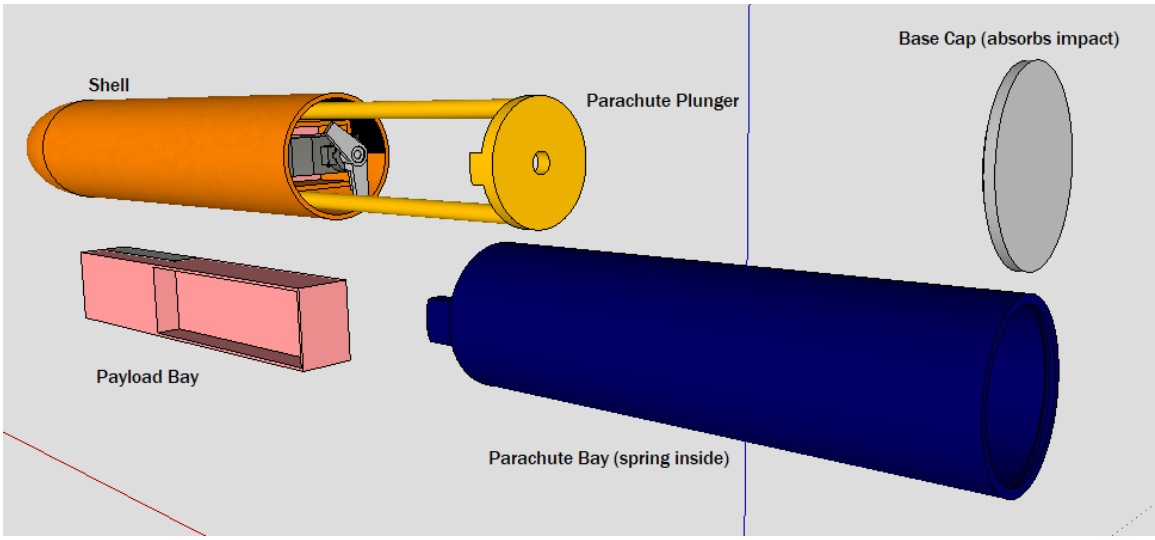

Figure 28. Prototype 3: Developed in SketchUp

Our next step was to design the behavior of the projectile. We used a development board to create the circuitry for the payload. The development board is depicted in Figure 29. We included an accelerometer, barometer, Arduino, VEmesh radio, and tilt sensor. We also included a micro servo in the payload, which is not depicted in Figure 29.

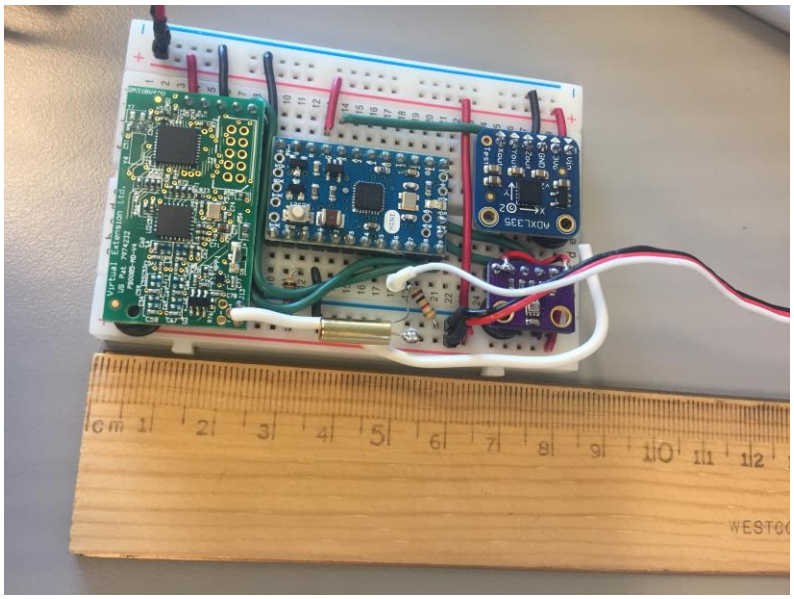

Figure 29. Prototype 1: Developed in SketchUp

Arduino provides an open source software called Arduino IDE which uses PYTHON language to create sketches. Sketches are compiled and then uploaded through the boot loader on the Arduino microprocessors. We wrote a sketch that included some feedback from the Arduino Pro Micro. We programmed the projectile to transmit an "INITIALIZED" message after start up. Upon being launched, the projectile transmitted a "LAUNCHED" message and began to feed a stream of flight characteristic messages back to the gateway. These flight characteristics included a time stamp, altitude, and acceleration. We used an accelerometer to sense the shot. Interestingly, we discovered that the acceleration spike occurred too quickly for the accelerometer to sense, so we had to adjust the frequency at which readings occurred. Shooting out of a gun, a projectile experiences instant acceleration and then graceful deceleration along the trajectory until impact. We added the altimeter's changes as a second set of criteria with which to sense the shot. The altimeter was already installed in order to sense apogee, the highest point along the projectile's trajectory. At apogee, we instructed the Arduino to send high voltage to the micro-servo, which let the spring expand and pushed the parachute out. With the parachute deployed, the Arduino transmitted a "DEPLOYED" message and began to act as a relay in the network. As a relay, the projectile could connect any other nodes in our experiment. We accepted that the relay mode would only be in effect during descent, even though that limited the total available time that command and control information could transfer between nodes.

After designing the projectile's behavior, we shifted back to our assembly line of effort. We removed the payload from the development board and soldered them for installation in Prototype 3. The soldered payload is shown in Figures 30 and 31.

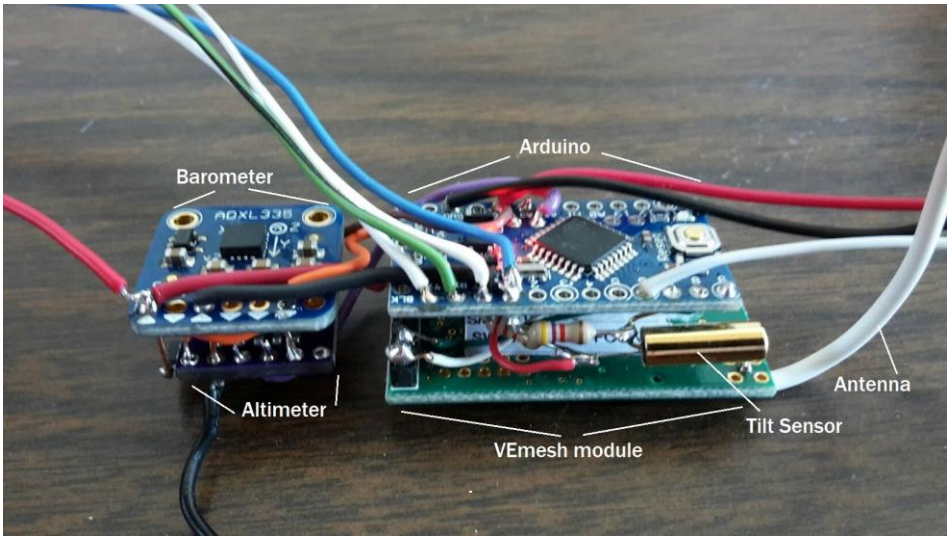

Figure 30. Prototype 3 Electronics

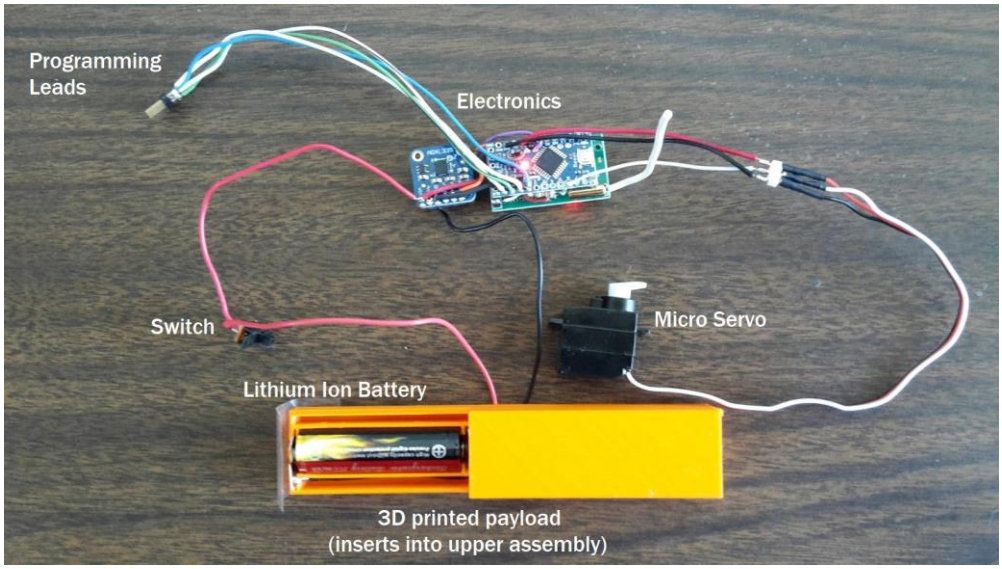

Figure 31. Prototype 3 Payload Assembly

We inserted the 3D printed payload into upper assembly and proceeded to test by launching Prototype 3 with the PLT Mini. Prototype 3 remained intact at 100% in-fill and we deemed it acceptable for inclusion in our first discovery experiment.

We noticed, however, that the altimeter was slow to sense apogee. Consequently, the parachute deployed late, limiting the relay mode duration. We added a tilt-sensor in an attempt to shorten the delay and maximize the projectile's controlled descent time

under the parachute's drag. Although the tilt-sensor worked flawlessly on the test bench, it acted unexpectedly in our first shots with the PLT. As the projectile left the barrel, the small ball inside the tilt sensor shot up the sensor barrel as it rebounded against the acceleration spike. The circuit opened in error and the parachute immediately deployed. We adjusted the sketch by logically attaching the tilt sensor to the Arduino only after the Arduino went into its "LAUNCHED" state. Our adjustment worked. We finished our sketch with instructions for the Arduino to transmit a "TOUCHDOWN" message when the projectile hit the ground. In those instructions, we told the Arduino to stop acting as a relay. Once on the ground, therefore, our network ceased to exist, although it could still function if programmed to do so. The Prototype 3 Arduino sketch is included for reference in Appendix B.

Finally, satisfied with the projectile's feedback functionality, we concluded our feasibility analysis. We proved Phase I's hypothesis by creating a working projectile by combining commercially-available electronic components with a 3D-printed assembly. Our working prototype introduced several functional constraints that would impact the design of our Phase II experiments. These functional constraints included the flight and signal range, time of flight, as well the duration of time that Prototype 3 would act in relay mode. The next section describes our Phase II design.

2. Phase II—Retrieving Data from a Remote Node

During Phase II experiments we planned to co-locate the projectile with the USB gateway and Windows interface. The gateway and pneumatic line thrower simulated the node with reachback. We placed a remote node beyond signal range. The remote node simulated its own cluster. The simulated remote cluster had a command and control message that needed to be relayed back to someone. The C2 message was simply "MISSION ACCOMPLISHED. ALL UNITS BACK TO BASE." Included in the remote node's Arduino sketch. Figure 32 shows the experiment topology during the second Phase.

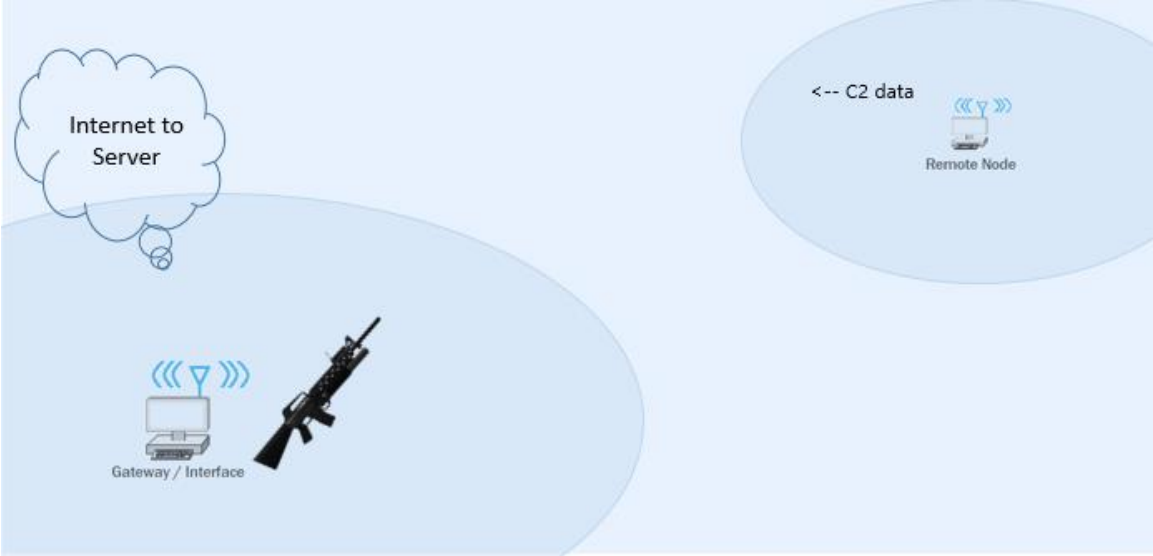

Figure 32. Phase II Experiment Topology

Although the clusters were not comprised of many nodes, we believed that they met our criteria. We knew that successfully transmitting C2 data in the first experiment would suffice as a proof of concept that a network could be created just for the short amount of time required at the application layer, then disestablished after the projectile impacted the ground. We planned to capture the throughput, the time of flight, and the range in order to make observations about the network behavior. We added the captured C2 data by instructing the gateway to post the received data to the CENETIX server. Phase III would test whether or not a cluster could take action on received C2 instructions over a bursty network. Phase III is discussed in the next subsection.

3. Phase III—Sending Data to a Remote Node

Phase III experiments also co-located the gateway and the pneumatic line thrower. Phase III would test whether or not a cluster could act on received C2 instructions over a bursty network. To meet our criteria, we used a remote UGV and sent it movement instructions. We changed the Arduino sketch to hold the movement instructions until after parachute deployment. If successful, we would observe our UGV executing a zig zag pattern. This simplified experiment topology did not require coordination with role players, nor did it require the added sophistication of other types of movement

instructions such as new grid coordinates. Figure 33 shows the experiment topology during Phase III.

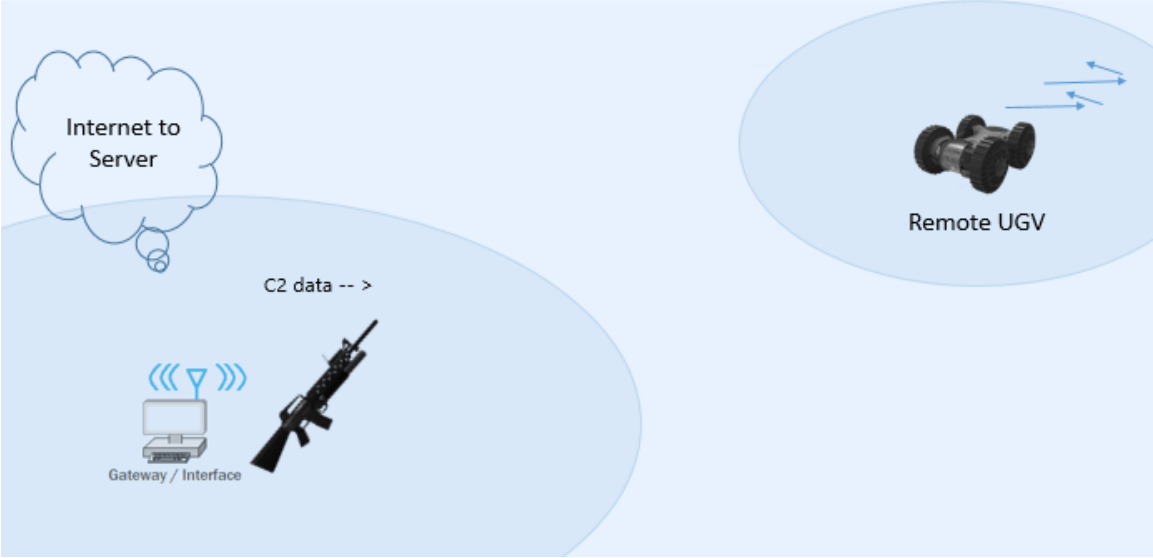

Figure 33. Phase III Experiment Topology

4. Phase IV—Scenario Vignettes

To conclude the campaign of experimentation, we created models of several potential tactical scenarios that employ a projectile-based node. These models were informed by data collected during the Phase II and Three experiments. These models represent a few hypothetical scenarios and are intended to illustrate the tactical applications for the reader. The amphibious raid vignette is available to watch at: https://youtu.be/k4xQExDC5l0. The unmanned undersea vehicle vignette is available to watch at: https://youtu.be/C0K4-1R8VB0.

THIS PAGE INTENTIONALLY LEFT BLANK

V. EXPERIMENT OBSERVATIONS AND ANALYSIS

This chapter provides experiment findings and our analysis. Observations made while carrying out the campaign of experimentation impacted the direction of follow-on experiments as we adjusted the prototypes and experiment design to test the bursty networking proof of concept. In order to capture the impact of observations made during experiment execution, we offer the Phase II and Three subsections as narratives followed by a list of observations by topic. At the end of the chapter we include an analysis. The analysis starts with a large perspective for the future use of bursty-networks in command and control of forces operating in EM-hostile conditions. The analysis concludes with an in depth look at what we believe to be desirable qualities of the short-living, highly mobile nodes in future bursty networks.

A. EXPERIMENT OBSERVATIONS

The following subsections record our observations during each phase of experimentation.

1. Phase I—Feasibility Analysis Observations

This section provides observations we collected during our feasibility analysis. Our hypothesis was that we could create our own suitable prototype for use in the final phases of our experiment campaign. During Phase I, we did successfully create a working prototype. Although our prototypes continued to evolve during Phases II and III, the observations provided in this section are solely from Phase I. Tables 2-4 lists our observations by prototype.

Table 2. Prototype 1 Observations

Component	Observation	Modification
50.8mm Spud Gun	Variable power each launch	Adopted PLT Mini and Restech Norway 230 models for launching follow-on prototypes
Rubber Band	Insufficent power to hold back spring	Switch from rubber band to micro-servo arm for spring retention in follow-on prototypes
Rubber Band Channels	Friction and hard corners occasionally caused the rubber band to become stuck, delaying the parachute deployment	Eliminate rubber band channels in follow-on prototypes
Time-Delay Sketch	Simple timer after power on produced unreliable deployment of parachute, either early or late during flight	Add components and program them to detect apogee in follow-on prototypes
Flight Characteristics	Stable.	None required.

Table 3. Prototype 2 Specific Observations

Component	Observation	Modification
Upper Assembly Side Door for Parachute	Side placement added air resistance during flight. Result was unstable flight path.	Design parachute to deploy from base of projectile in follow-on prototypes
Micro servo holding Bottom of Door	Internal spring resistance created a protrusion at the top of the door, exacerbated by air flow during flight	Design parachute to deploy from base of projectile in follow-on prototypes
Electrical components in lower assembly	Placing electrical components nearer to the impact of the launcher is not desirable due to greater force at launch.	Design internal payload in upper assembly of follow-on prototypes
Flight Characteristics	Lack of stability during flight	Design follow-on prototypes with better balance, both in regards to the center of the cylinder and forward to aft.

Table 4. Prototype 3 Observations

Component	Observation	Modification
Base Cap	Cap designed to absorb impact at launch and be deployed with parachute was routinely destroyed by force at launch	Design follow-on prototypes with more robust area to absorb initial impact at launch
Micro Servo Arm	Prototype 3 had two shafts that retained the spring and plunger in the parachute bay. These shafts put pressure on the micro servo's arms that the torque could not release at deployment	Design follow-on prototypes with a trigger mechanism. Micro Servo has sufficient torque to pull trigger, but not to retain shafts.
Lower and Upper Assembly Join	The join at the lower and upper assembly proved to be a weak point in the design. We compensated for its propensity to break by taping over it with duct tape.	Design follow-on prototypes with a single assembly.
Assembling while Loaded	After loading the spring and inserting the parachute, putting together the projectile's upper and lower assembly was tedious.	Design follow-on prototype with an ease of loading in mind.
Flight Characteristics	Flight path was true. No observed wobble during flight.	None required.

2. Phase II—Remote Node Experiment Observations

We conducted several incremental Phase II experiments from October 26 to October 29, 2016. Before departing for the range on the 27th, we tested Prototype 3's functionality by hand. At the range, we relied on the Restech Norway PLT mini launcher. The PLT mini launcher is shown in Figure 34. We chose not shoot the Restech Norway 230, which releases a greater amount of pressurized air for added range. We shot one Prototype 2 and three Prototype 3s on the range during the first experiment. Prototype 2 did not have a payload. We shot it for the sole purpose of confirming flight characteristics. Prototype 2's base module, designed to hold the payload, shattered under the force of the PLT mini on launch. The parachute bay, nose, and what was left of the

base flew in a wobbly manner, which confirmed our suspicions about its lack of flight worthiness.

The first Prototype 3, printed at 22% in-fill, also shattered the bay closest to the base of the PLT mini. Figure 34 captures the first Prototype 3 launch.

Figure 34. Prototype Breaking On Launch

The remaining part of the first Prototype 3, which is the orange plastic in Figure 34, flew approximately 50 meters high and 100 meters down range. A strong tail wind carried it significantly. Remarkably, the payload itself survived the shot. In fact, we were able to use the same payload bay until shooting on October 29th.

Concerned about losing all of our prototypes on the first shot, we decided to suspend the projectile in the barrel using some duct tape. We wrapped the duct tape around the intersection of the payload bay and the parachute bay, thinking that placing it

there would add stability. The duct tape kept the projectile approximately 4 inches above the internal base of the launcher. The tape is visible in black in Figure 35.

Figure 35. Prototype Ready to Launch

Our second shot with Prototype 3 remained intact throughout flight, but the parachute did not deploy. Figure 36 captures the trapped parachute. Time of flight was roughly six seconds without the parachute functioning. Zooming in on the captured image in Figure 36, the tail looks slightly wider than the nose of the projectile. This observation led us to believe that the parachute came partially out, but was unable to fully deploy. As recorded on Observer's Notepad on 27 October in Figure 37, the projectile successfully initialized, launched, deployed, and then initialized again after landing. During the second trial, with Prototype 3, also successfully transmitted part of the C2 message from the remote node.

Figure 36. Prototype Passing Apogee

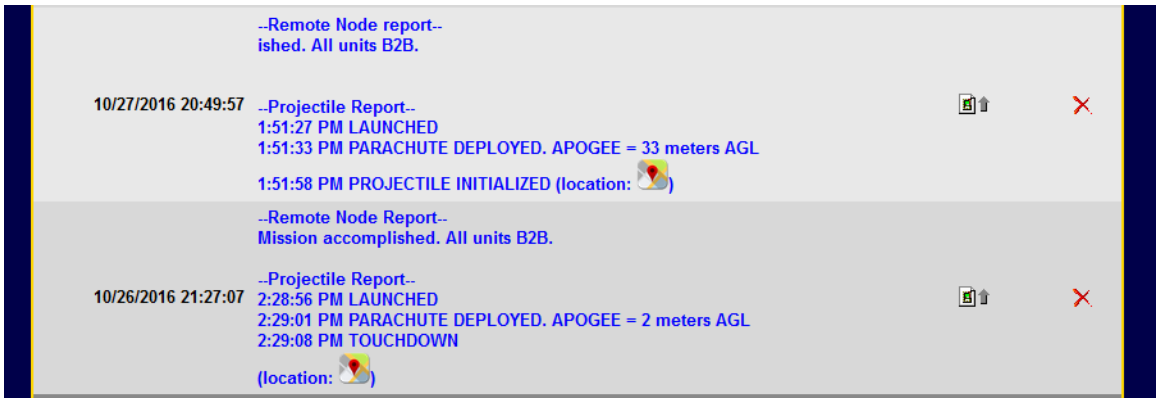

Figure 37. Observer Notepad Record of Data from Remote Node

Our third and subsequent shots with Prototype 3 remained intact throughout flight, but the parachute proved unreliable. After seven more shots, our remaining Prototype 3s had cracked at the base and at the intersection between the payload bay and the parachute bay. We also damaged the micro servos. We decided to shoot the projectile a final time without a parachute bay. The shot without a parachute bay was not successful. The shot flew in a sideways spin and landed hard. Upon recovery, we noticed that the battery had came through the PLA wall of the payload insert, pushing the

electronics out of alignment, and ending the possibility of more trials during the first experiment phase. Figure 38 shows the damaged payload insert of Prototype 3.

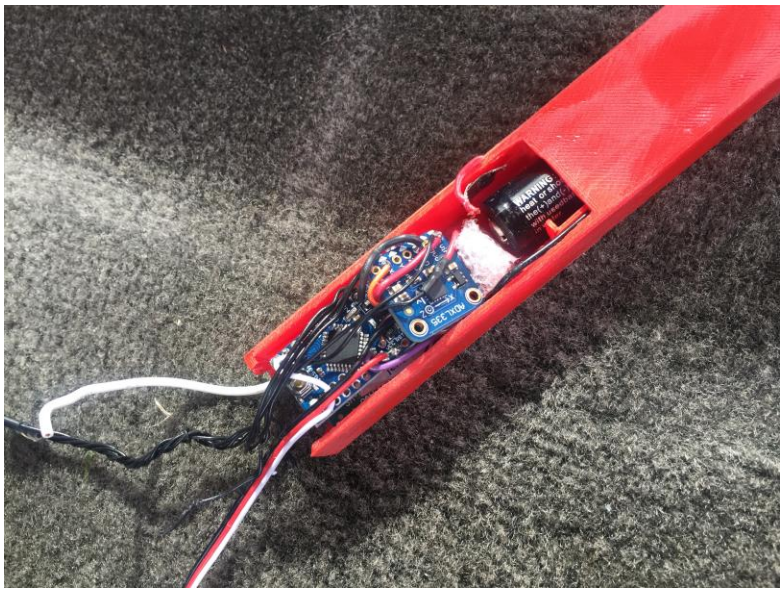

Figure 38. Payload Damage

In multiple trials we made a few observations that led us to modify our projectile design before proceeding with Phase III. Prototype-specific observations are included in Table 5. Given our required changes, we decided to adopt a larger prototype incorporate the already-existing hard plastic base attached to the projectile included with the PLT mini. We scaled up the projectile so that we could eventually incorporate a different mesh radio. We wanted to use Persistent Systems' Wave Relay MPU-4 radios so we made our projectile Prototype 4 large enough to hold the entire circuit board from the MPU-4. For our Phase II experiments, we determined that we would need a trigger mechanism for the micro servo to pull in for the prototype.

Table 5. Phase II Prototype-Specific Observations

Component	Observation	Modification
Micro Servo	Insufficient torque to release parachute spring	Upgrade Micro Servo Add trigger insead of direct pressure
PLA plastic	Brittle to impact and pressure	Change materials for exterior
PLT Mini	Insufficient range	Upgrade to Restech Norway 230
Parachute Bay	Created too much surface tension for the parachute to slide through	Expand the parachute bay Change to smoother materials

Observations about the network functionality and behavior are included in Table 6. Phase II experiments successfully connected two clusters together using a 3D-printed projectile for a duration required for one C2 message to be transmitted from a remote node to the gateway.

Table 6. Phase II Network Behavior Observations

Element	Specification	Result
C2 message	Transmit short message from the remote node to the gateway	Success: transmitted a short message
Network Life	Defined as the period of time that the two clusters were interconnected.	Six seconds on average, through all Phase II experiment trials.
RS 232 Protocol	Limits number of total bits, as it fragments a given message into many small parts and has a definite maximum transmission limit.	Success: Our reduced message size was able to be transferred after network established.
Baud Rate	2400 bps selected on the VEmesh interface prior to the trials.	Success: No errors noted on Observer's Notepad.
Interference	Monitor network behavior for existence of interference.	None noted during experiments.

Given the number of projectile modifications required for Phase III experiments, we planned our Phase III experiments as a part of the larger CENETIX experiments and multi-domain network architecture at Camp Roberts, CA during the third week of March, 2017. The CENETIX multi-domain architecture is depicted in Figure 39.

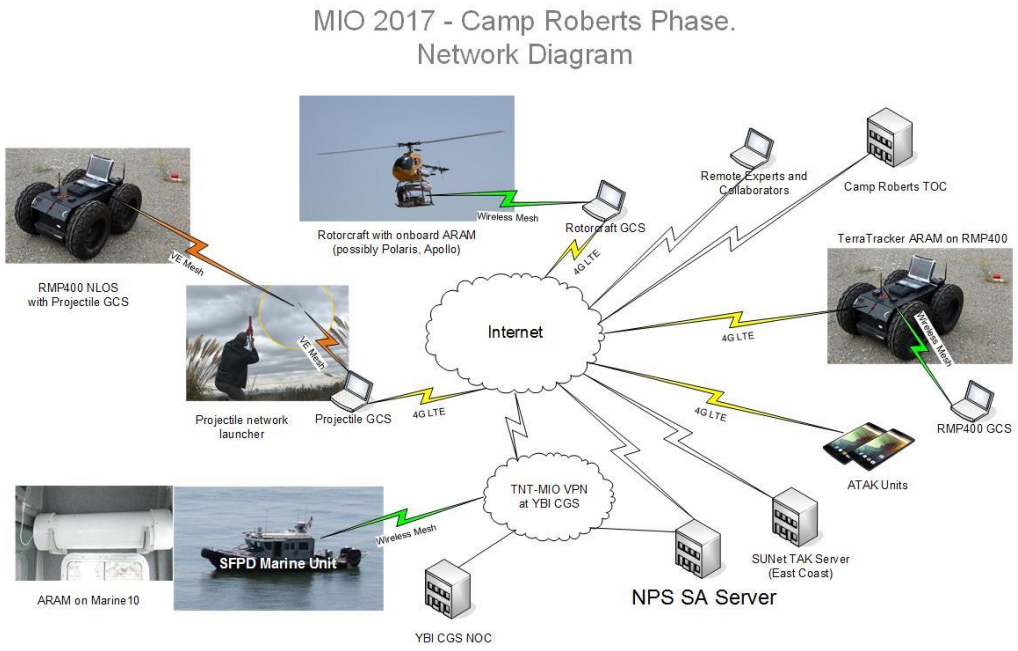

Figure 39. CENETIX Maritime Interdiction Operation (MIO) 2017—Camp Roberts Phase Network Diagram

The following section provides a narrative and observation table for our Phase III experiments at Camp Roberts in March, 2017.

3. Phase III—Command and Control Message to Remote Nodes

Phase III experiments were advanced discovery experiments designed to test whether or not a cluster could act on received C2 instructions over a bursty network. We conducted six trials during Phase III. To achieve our criteria, we used a remote UGV and sent it movement instructions. We modified the Arduino sketch to hold the movement instructions until after parachute deployment. If successful, we would observe our UGV executing a zig zag pattern. The prototype also evolved for Phase III experiments. Those changes are recorded in the following paragraphs, followed by a narrative of the conduct of Phase III.

As stated in the previous section, we upgraded several systems from Prototype 3 for Prototype 4. We included a micro-servo with more torque and designed a trigger mechanism for Prototype 4. To help redesign the deployment mechanism, we searched

for existing examples. The water rocket community had a variety of options available. After researching home-rocket options, we modeled our new deployment mechanism design after Air Command Water Rockets' Shadow Model (Air Command, n.d.). Figure 40 shows Prototype 4's trigger mechanism comprised of aluminum parts. As depicted in Figure 40, the micro-servo in Prototype 4 did not retain the plunger like the micro-servo in Prototype 3. While demanding extra space, incorporating the trigger mechanism improved the probability that the parachute would consistently deploy.

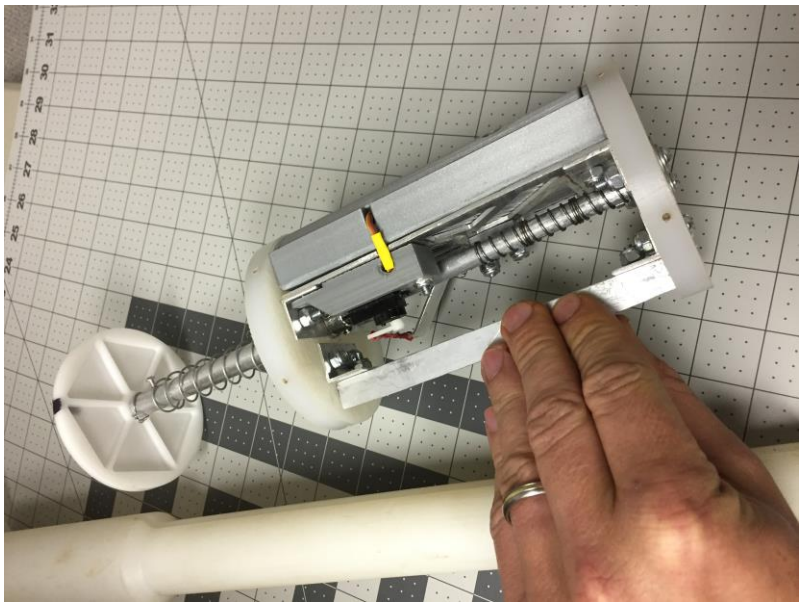

Figure 40. Prototype 4 Payload Assembly

While we planned to use the extra space for the MPU-4 radio when the parachute proved its reliability, we used the extra space to install the entire development board into the payload bay. Figure 41 shows the development board inside Prototype 4.

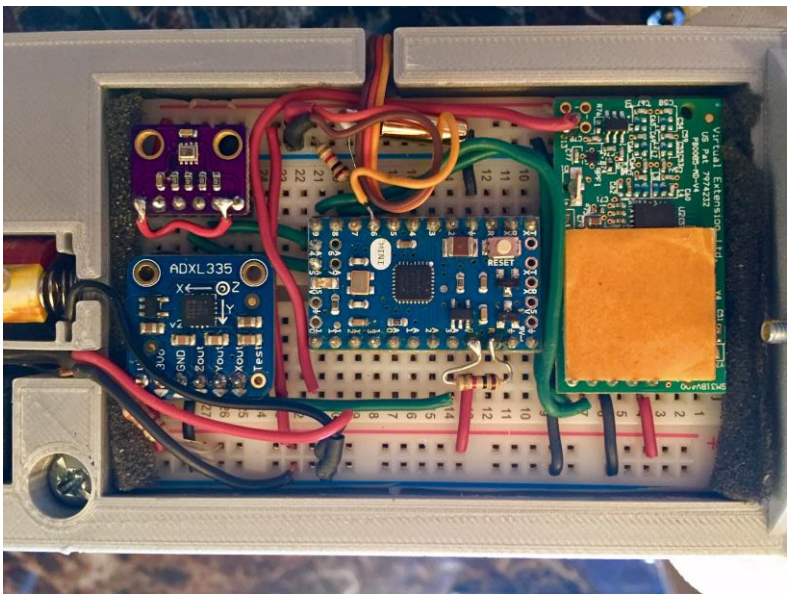

Figure 41. Development Board Inside Prototype 4

We also discarded the 3D-printed body in Prototype 4. We still used 3D-printed PLA parts for some internal parts, but we opted for a fiberglass cylinder for Prototype 4's shell, aluminum for the internal structure, and milled polyethylene for major internal surfaces. We used a 3D Carbide Nomad 883 computer-numerical control (CNC) mill to mill the polyethylene and aluminum to specification. The Nomad 883 proved to be a great asset. Provided by the Robodojo, it allowed us to accurately and quickly manufacture robust parts. Additionally, we also borrowed a cylindrical shaft from a projectile that came with the Restech Norway 230 in order to successfully absorb more impact upon launch. The shaft was also made of polyethylene. Figure 42 shows Prototype 4 with Restech Norway's PLT mini.

Figure 42. Prototype 4 With PLT Mini

Our Prototype 4 design relied on the trigger assembly to separate the borrowed shaft from the payload bay at apogee and also to push the parachute out for deployment. We selected springs capable of creating adequate force. The trade-off for creating a bigger Prototype 4 with bigger springs and more impact-resistant materials was the projectile's weight. Prototype 4 weighed nearly 3 lbs. Restech Norway's PLT mini did not produce the range we required for Phase III experiments, so we adopted the Restech Norway 230. We also added a larger dive tank to the 230 model launcher. We added the larger dive tank to increase the number of shots possible without refill. Camp Roberts, our test site, is not located near facilities that can re-charge tanks. Because the pneumatic line throwers are a surrogate for traditional weapon systems, the modification did not affect our criteria space. Figure 43 shows the modified model 230 launcher.

Figure 43. Rescue 230 Launcher with DIN-Style Dive Tank

In preparation for Phase III experiments at Camp Roberts, we equipped the Segway-based UGV with a remote node. The remote node was comprised of another Arduino microprocessor, VEmesh module, and power supply. The remote node's purpose was to receive the data payload from the gateway computer and instruct the UGV to execute movements. The UGV is depicted in Figure 44.

Figure 44. Unloading UGV at Camp Roberts

Once we arrived at Camp Roberts, we located an area that was suited for our experiment. We chose a lightly used road and a empty field. The field and the road were separated by a berm. The berm was approximately two meters high. The berm's heighth was sufficient to block the line-of-sight connection from the gateway to the UGV. We placed the UGV on the road, and drove with the gateway across the field. There we checked that the UGV and the gateway were not able to communicate directly with one another. Figure 45 depicts the Restech Norway 230 and Prototype 4 just before launch.

Figure 45. Prototype 4 at Camp Roberts

We executed six nearly identical trials in Phase III at Camp Roberts. During the first trial, the Arduino never sent a "LAUNCHED" message back to the gateway and the user interface. Since Arduino did not detect conditions for deployment, the projectile never sent C2 information to UGV. Prototype 4's payload did not separate from the lower assembly. The prototype landed hard, with the lower assembly and payload still connected, all without the parachute deploying. After retrieving the projectile, we noticed a battery fault. The battery emitted no power. We suspected that the launch created the battery fault, which prevented the projectile from behaving according to our design. However, we could not rule out the possibility that the Arduino and VEmesh module did have power during flight and that our payload failed to sense the launch. We replaced the battery and bench tested the payload's behavior. It behaved according to our design. Satisfied with our modifications, we proceeded to the second trial.

The second trial exhibited the same behavior as the first experiment. The parachute did not deploy. Possible reasons included: accelerometer malfunction, barometer malfunction, reset sensor triggered during flight, or battery failure. We also noted that it took extra torque to separate the lower assembly from the payload after recovering Prototype 4 from the hard landing. We suspected that the spring and trigger inside the payload would be insufficient to create separation in future experiments. To

compensate for the torque necessary to separate the lower assembly from the payload, we used the recovery rope included with the Restech Norway kit and fastened it to the lower assembly. The idea behind the rope was that it would run out at apogee and then jerk the lower assembly away from the payload. Separated from the lower assembly, the payload would then be free to deploy the parachute. Figure 45 (above) shows Prototype 4 attached to the rope provided in the Restech Norway kit.

We replaced the battery again for the third trial. While Prototype 4 did not lose power in the third trial, it failed to transmit a "DEPLOYED" message and begin to give C2 instructions to the UGV. Several seconds after landing, the UGV conducted maneuvers which indicated that the data was received by the remote node. The C2 instructions were likely transmitted because our program sketch included a reset to "INITIALIZED" after 30 seconds.

Trials four through six produced similar results to trial three. The parachute did not deploy and a short time after Prototype 4 landed, the UGV conducted its maneuvers. After producing consistent results, we decided that we needed to rethink our prototype and experiment model. We recorded our network behavior observations, which are provided in table 7.

Table 7. Phase III, Protoype 4, Network Behavior Observations

Element	**Specification**	**Result**
C2 message	Transmit short message from the gateway to the UGV	Success: transmitted a short message, UGV acted on instruction only after projectile was on the ground
Network Life	Defined as the period of time that the two clusters were interconnected.	Unable to measure projectile never sensed and reported status during flight duration.
Interference	Monitor network behavior for existence of interference.	None noted during experiments.

The following paragraph describes the analysis that drove further prototype revisions. Table 8 contains a summary of the analysis.

Table 8. Phase III, Prototype 4 Specific Observations

Component	**Observation**	**Modification**
Descent Mechanism	Unreliable performance	Remove from follow-on prototypes
Power supply	2 of 7 trials killed the battery. This is likely because of the added impact at launch due to increased size and weight of Prototype 4	Reduce size and weight of follow-on prototypes
Internal Components	Removing the descent mechanism provides the opportunity to remove most internal components: microprocessor, accelerometer, barometer, and tilt sensor	Remove all unnecessary components from follow-on prototypes.

We included a parachute in all previous prototypes for two reasons: in order to increase flight time and in order to protect the payload. However, our Phase II and Three experiments proved that the payload remains intact after landing on grass and that the flight time without the parachute functioning provides sufficient opportunity for a message to be transmitted across the network. We therefore decided to simplify our prototype for future experiments. We decided to remove the parachute and simplify the program sketch. With the parachute removed, we were also able to eliminate the accelerometer, barometer, tilt sensor, parachute, plunger, springs, servo, and lever. With many of the components removed, we decided we did not require an Arduino either. We therefore designed Prototype 5 as a VEmesh module powered by the battery inside a small 3D printed assembly. Prototype 5 is depicted in Figure 46.

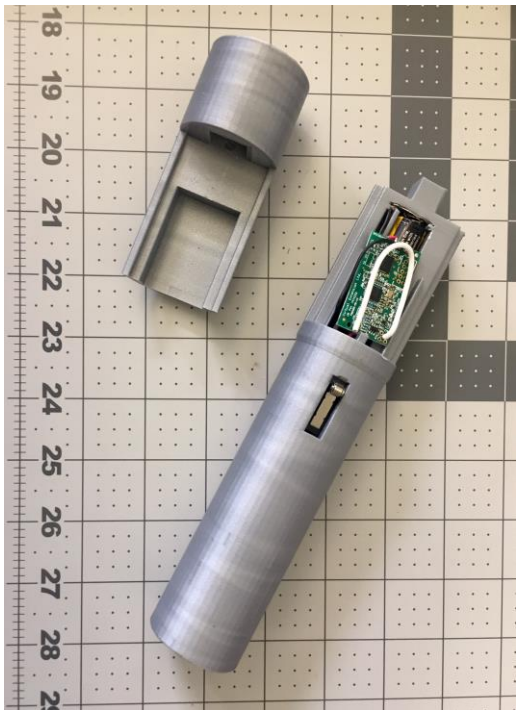

Figure 46. Prototype 5

We designed Prototype 5 to turn on only after launch. The small roller switch that provided on-off functionality is visible in Figure 46. Prototype 5 would power on, join the network, and then act solely as a relay for the duration of flight. Prototype's simple design promised to produce fewer points of failure. However, upon testing Prototype 5, we discovered some surprising and important network behaviors. The following paragraphs describe the bench testing and our network behavior observations. Table 9 summarizes the network behavior.

Table 9. Phase III, Prototype 5, Network Behavior Observations

Element	Specification	Result
VEmesh module Power (Layer 1)	Virtual Extension introduces power to the VEmesh module in a progressive manner upon start up, from no power to full power over a given time.	VEmesh module components are better protected by the slow power on process. However, time required for network to converge is extended
Pseudo-Random Hops (Layer 2)	Virtual Extension's Frequency Hopping Spread Spectrum Approach uses a default 21 frequencies to hop across using a pseudo-random code. Value can be changed from 51 to 1.	Network converged in 14 seconds with 21 frequency default. Network converged in 8 seconds when hopping on just one frequency.

We designed a ground-based test how our network would behave during Prototype 5's flight. We programmed the VEmesh user interface to produce an audible beep whenever it received data from the remote node. Then we relocated the remote node from the VEmesh gateway so that the audible beeping stopped. Physically, the beeping stopped at around 100 meters with line-of-sight separation. We placed Prototype 5 within LOS of both the remote node and the VEmesh gateway. Then, with a timer, we measured the time from switching on Prototype 5's power to the first audible beeps at the VEmesh user interface. Figure 47 shows the ground test design.

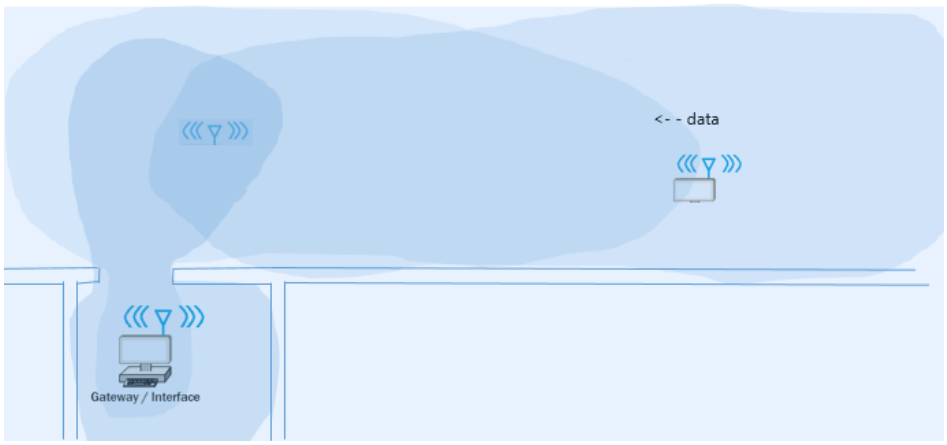

Figure 47. Prototype 5 Ground Test Design

In all of our tests, audible beeping began at an average of 13 seconds after sending power to Prototype 5. Since the flight time for our projectiles was previously 6–8 seconds without the parachute deploying, we knew we had a problem if we wanted to continue to limit our transmission to the duration of flight. We wondered how we were able to transmit data inside of the flight windows during the previous phases of our experiment. New variables in our experiment included delaying powering on the projectile until launch and excluding the Arduino microprocessor. We suspected that network activity occurred in previous models as we prepared to fire. Specifically, the projectile's VEmesh module acquired the network while in proximity with the VEmesh gateway before launch. Conversely, when powering on after launch, we ensured that all the network traffic concerned with the OSI model's layers was accounted for during flight. Further, we suspected that the delayed network convergence was due to protocol activity at layers 2 and 3 in the OSI model. Specifically, we thought that the FHSS that enables Simulcast was the root cause of the additional delay. We needed to find out how a node just joining the network receives the pseudo-random code and the timing that the network is hopping across.

We contacted Virtual Extension and learned that our VEmesh radios used a default 21 different frequencies to hop between and that the number of frequencies was programmable. Virtual Extension allows the network programmer to use as many as 50 frequencies or as few as one. We chose to reduce the number of frequencies to one. With the VEmesh gateway and all nodes programmed to use a single frequency, we returned to the test illustrated in Figure 47. We suspected that we would be able to significantly reduce the delay from the original 13 seconds. To our surprise, however, the audible beeping still began at an average of 8 seconds after sending power to Prototype 5.

We again contacted Virtual Extension. This time we learned that Virtual Extension uses hardware that slowly introduces power over the circuit upon start. By slowing the introduction of power, Virtual Extension decreases the risk of burning a component on the circuit. Since the VEmesh product we used was originally designed as a mesh network of remote sensors in agricultural scenarios, the slight delay in acquiring the network was never a complaint from users. For agricultural users, slow power increases

the likelihood that each VEmesh module lasts longer, which is a desirable feature. However, for our purpose of use inside a short-living node, the delay becomes unacceptable. We had several options. We could power Prototype 5 before launch, we could adjust our design variables and search for longer-living nodes, or we could allow Prototype 5 to relay from the ground after landing. After deliberation, we decided that we had observed sufficient criteria for our study to make observations. Thus, we decided to conclude our campaign of experimentation.

B. ANALYSIS

This section provides a broad-perspective analysis of the campaign of experiment's results. We begin the analysis with our own working model that drives the need for short-living networks. This model is created using the systems theory framework discussed in Chapter III. We follow the model with an analysis of the desirable features of a short-living node within a bursty network. We use the OSI model discussed in Chapter III to logically categorize our desirable features. We then conclude the analysis with a network-level perspective on functionality and security. We use the CIA triad from Chapter III as a rubric for the functionality and security analysis.

1. A Model for Operating Short-Living Networks

Our campaign of experiments explored the proposition that a force can conduct command and control by using bursts in a short-living network. We considered how a force might actually employ bursts during a tactical operation. We also considered the influences affecting a force's decision to emit signals. The need to coordinate seems to balance with probability of EM detection in a traditional *cat and mouse game*. Figure 48 is a systems theory model for operating short-living networks under different EM-hostile conditions.

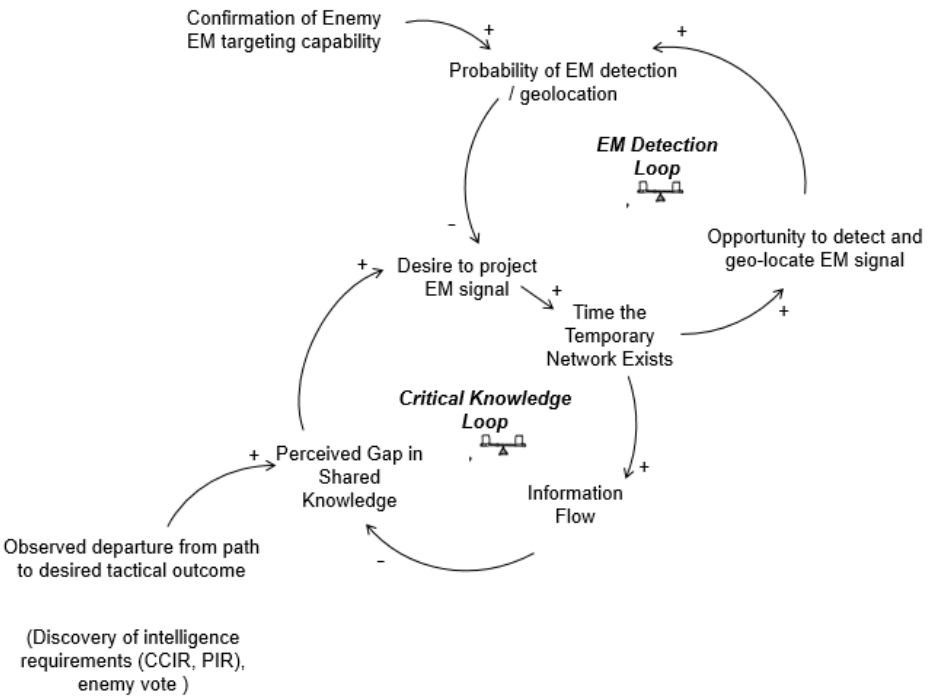

Figure 48. A Model for Operating Short-Living Networks

From a systems theory perspective, we believe that a force's critical communication needs can serve as a feedback loop that drives the physical existence of the network itself, and in doing so, provide passive defensive countermeasures while the force operates in an EM-hostile environment. We created a self-balancing system to illustrate how communication requirements during a tactical scenario are offset by the probability that the force will be targeted through those communications.

Our model for disruption-based networking functions within a self-balancing system with two interacting feedback loops. In Figure 48's critical knowledge loop, a tactical force maintains a perception of shared knowledge. For instance, imagine a company that conducts a final brief and rehearsal with the commander. The platoons depart to begin the operation. Each platoon perceives that they are "on the same page" as the other platoons. Each platoon has some conception of how the mission should go if their company is to be successful. Now, imagine that one of the platoons observes a departure from how the mission is supposed to go. There may be an ambush by the enemy or some civilians where none were expected. That platoon perceives a gap in

shared knowledge with the company and has a greater desire to communicate over the network. Likewise, if there is no ambush or no civilians, the platoon does not perceive a change in shared knowledge and does not believe there exists a need to communicate. That is how the Critical Knowledge Loop works in Figure 48. The perceived gap in shared knowledge influences the force's desire to establish a short-living network. If the team believes the shared knowledge gap is great or urgent, they will feel a great need to shoot the projectile and transmit critical data in order to close the shared knowledge gap. When the network is created, information flows and the gap in shared knowledge closes. The force "gets on the same page."

The observed departure from the path to the desired tactical outcome injects into the critical knowledge loop in Figure 48. The ambush scenario illustrates one of these injects. Intelligence requirement discovery may also serve as observable criteria in a tactical scenario experiment. Intelligence requirements (IR) are defined as requirements for intelligence to fill gaps in the command's knowledge and understanding of the battlespace or enemy forces (Joint Chiefs of Staff [JCS], 2010). IRs are typically missing pieces of information about the enemy that a commander needs to know to make a sound decision. IRs can be classified as commanders' critical intelligence requirements (CCIR) and priority information requirements (PIR), which are those IRs deemed critical to facilitating timely decision making and IRs stated as a priority for intelligence support respectively.

The EM Detection Loop in Figure 48 is a counter-balancing influence in our model. The EM Detection Loop is connected to the Critical Knowledge Loop by the network existence time element. That is, the time that the temporary network emits detectable signals gives the adversary more opportunities to detect and target the platoons in the scenario. The platoons in the company understand that direct correlation between network existence and the probability that they will be targeted. There is another significant element in the EM Detection Loop, though. That significant element is the company's perception of the risk of detection and targeting. Perception of detection risk depends on confirming information about the enemy's EM capability. For instance, if the enemy is not known to be able to target using the EM spectrum, the company is free to

emit any signal they want. Persistent signals networks used in previous DOD operations are an example of a negligible force from EM detection loop.

If the enemy is known to be adept at EM detection and geolocation, the force will perceive that detection is probable. When EM detection is possible or probable, the EM detection loop activates. The force from the EM detection loop changes the equation for network creation, dissuading the force from creating the network and communicating freely. No force wants to be targeted, so the force will only create the network until the knowledge gap becomes absolutely critical.

The self-balancing model is simply a model we've developed based on our understanding for the driving factors for bursty networks in tactical scenarios. Its purpose is to explain both why bursty networking may become necessary and how a force will balance network creation versus targeting by the adversary. In the next section, we provide our analysis on desirable qualities of short-living nodes using the OSI model.

C. OPEN SYSTEMS INTERCONNECT (OSI) ANALYSIS

The following subsections organize our findings by layer of the OSI model. Each layer will include a description of the protocols exhibited in our experiments and then some recommended protocols for inclusion in future experiments. In order to give those recommendations, we need to identify our underlying assumptions about operating bursty networks in the future operating environment. The next paragraphs describe those assumptions.

We make three major underlying assumptions about operating communication networks in the future operating environment. The first major underlying assumption is that persistent signals are more likely to be detected by adversaries. Furthermore, those adversaries will be able to geo-locate and target our forces using the signals that they are able to detect. The second major assumption is that strict emission control, also known as *going radio silent*, will be a sub-optimal option because the clusters need to act in a coordinated manner in order to produce mission success.

The third major underlying assumption is the most technical. We believe that end-to-end connections communication links across disparate clusters will not be feasible to maintain persistently during a single burst of the network. We believe this to be true of the network in a gamut of scenarios: naval warfare scenario in the littorals, in urban combat, or in reconnaissance missions. In fact, we believe that an end-to-end connection may not be possible to maintain within between nodes within a single cluster for long enough to support a communication session. Figure 49 illustrates these assumptions with a simple diagram clusters A, B, and C. The nodes that are within signal range of each other are connected by a dotted line. As Figure 49 shows, the upper left-most node in cluster A can reach the most remote nodes in cluster B, but no link is established to cluster C.

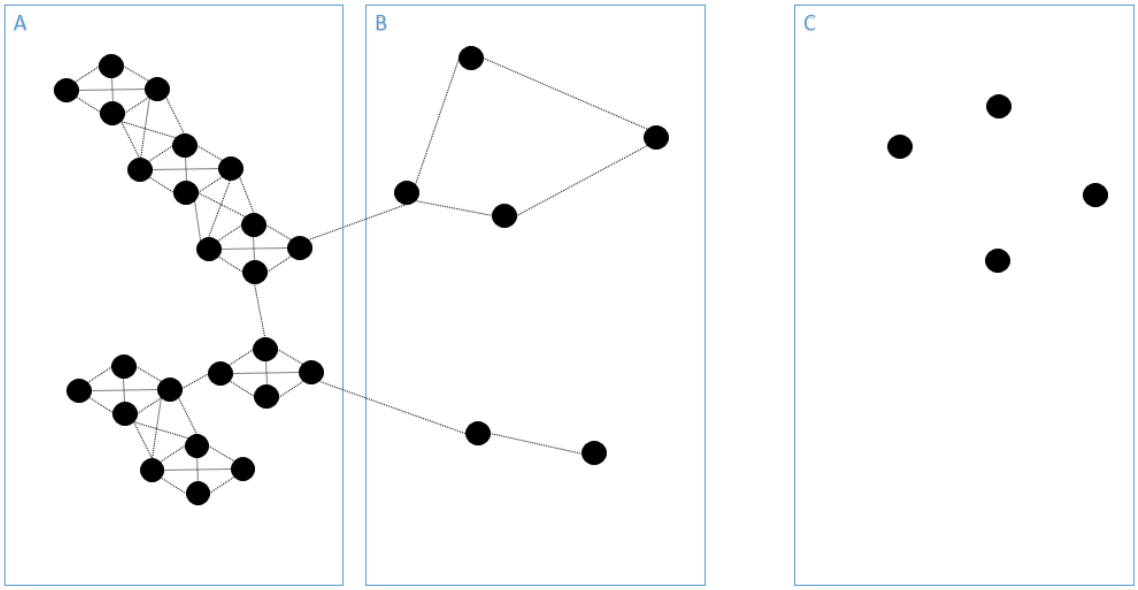

Figure 49. Clusters A, B, and C

In Figure 49, intermediate clusters B can not be relied upon to maintain an end-to-end connection between clusters A and C in figure 49. There are several reasons why. The first reason is that the mobility of the nodes within the clusters will cause frequent network topology changes that will not be mapped by clusters A, B, or C without network messages. Those network messages are not possible without transmitting a

signal that is detectable by an adversary. The second reason is that we believe that signals that are powerful enough to reach from cluster A all the way to the cluster C, even in a burst, increase the risk of an adversary detect the signal. Clusters A and C may be separated by 100 meters or 100 kilometers. Additionally, there may be many intermediate cluster Bs between sending cluster A and receiving cluster C. Each intermediate cluster B exercises various levels of mobility that would be strictly limited if they participate in an end-to-end communication session between clusters A and C. Active participation by intermediate nodes and clusters results in more signals being emitted, which also increases the risk that an adversary will detect and target our forces.

To demonstrate our assumptions, imagine an amphibious raid scenario where a naval force launches elements of the Marine Expeditionary Unit (MEU) into an urban area. Imagine that the adversary has proven an ability to target through EM detection, and that intelligence believes that the adversary can geo-locate signals if they persist for more than 30 seconds. Figure 49's clusters A in our imaginary scenario is the raid element of two Marine platoons which are represented as four Marine squads. Cluster B could be the rotary-wing assets providing them lift from ship to landing zone. If there were more than one Cluster Bs, suppose that those are comprised of patrol craft and UAV clusters. Cluster C consists the ships of the amphibious readiness group (ARG)/MEU, including the Landing Force Operations Center (LFOC). Applying our assumptions to the imaginary scenario means that: 1) the Marine platoons are out of communications range with the LFOC unless they are serviced by intermediate cluster Bs, and 2) that intermediate clusters of rotary-wing lift, UAVs, and patrol craft will be moving as dictated by mission and force protection requirements, which result in different possible end-to-end links at any point in the operation.

The following subsections detail the protocols in our experiments and provide our recommended protocols for inclusion in future experiments given our assumptions. Readers who require a more detailed description of the OSI model should refer to Chapter III.

1. Physical Layer

The physical layer properties were of primary importance to this study, because protocols at all higher layers are translated into physical layer signals. Activity at higher layers that allow users to communicate over the network become a physical signal that an adept adversary can detect. We measured signal range at the physical layer but connection time at the application layer. Because we knew that our VEmesh radios were emitting signals as soon as they were powered, we did not need to devise additional means to measure total physical emission time that included every layer of network traffic. Our VEmesh modules were equipped with monopole antennas which produce the effect of an omni-directional signal. Equipped with low power, our range was less than 200 meters.

In a tactical context, an adversary with a receiver will be unable to detect a friendly signal if the signal is indiscernible from the background noise. The transmitter's emission pattern and signal power are therefore major factors in detection. Outside of signal range, any receiver receives only what appears to be background noise. Omni-directional antennas provide the same signal in nearly all directions while directional antennas emit the signal in a more focused trajectory. Adversaries can position receivers anywhere within proximity of omni-directional antennas in order to detect friendly signals, while they have to be either in the pattern of a directional antenna or within one of the typical side-lobes. There is a trade-off in antenna choice. The more directional an antenna, the more aimed it must be in order to establish a successful connection. Aiming requires a priori knowledge of the location of the recipient. That knowledge will typically require that mobile nodes communicate their positions frequently, which may not be feasible or desireable under EM-hostile conditions.

Another physical layer consideration is data encoding. Selecting an appropriate data encoding technique is often a calculated trade-off between maximizing data throughput and finding an acceptable error rates in a given environment. We did not capture the encoding technique used with the VEmesh radios in our experiments. We did set bits per second to 2400, which was the maximum value offered in the Virtual

Extension interface. In future experiments, we recommend the continued balance between acceptable error rate, range, and maximum throughput.

2. Data Link Layer

Our VEmesh radios used a channelization protocol, FHSS. Channelization protocols generally require greater awareness of all nodes in the network in order to distribute the resources appropriately. In our experiments, a VEmesh module joining the network listened for the sequence of pseudo-random hops. Convergence time was approximately 13 seconds using the pre-set 22 number of frequencies. We reduced convergence time to 8 seconds by eliminating all hops. The additional overhead required to operate FHSS does reduce the routing message overhead at layer 3. VEmesh uses FHSS to drive its Simulcast routing at layer 3. The interaction between layers is an interesting point. In the VEmesh example, additional layer 2 message traffic virtually eliminates all required traffic at layer 3. This relationship between layers can be exploited if the tactical conditions permit. The perceived latency due to layer 2 message traffic may be able to be bypassed if conditions allow for all nodes to logically join the network before beginning the tactical scenario. As we demonstrated in our Phase II experiments, the nodes experienced virtually no delay in passing application layer traffic because they synchronized on the pseudo-random code and gained the timing from the VEmesh Gateway prior to shooting the projectile and placing the remote node. Applied to a tactical scenario, this means that if all nodes can synchronize before entering the range in which the adversary can detect using EM tools, sub-application layer message traffic can be reduced.

We recommend exploring random access protocols in future experiments with distrupted tactical networks. Carrier Sensing Multiple Access (CSMA) – Collision Avoidance (CA) is a nice starting point. Random access protocols seem well-suited for short-living networks because the alternatives, controlled access and channelized protocols, require an layer 2 overhead that will be challenging to support under EM hostile conditions. The frequent network mapping messages will require many more bursts, which adds to the likelihood of detection. Layer 3 is discussed in the next section.

3. Network Layer

Our VEmesh radios employed Simulcast to perform the routing function resident at layer 3. The technical operation of Simulcast is discussed in depth in Chapter IV. The results of employing Simulcast in the network behavior are important to note. Simulcast routing, because of its unique channelization of time and frequency at layer 2, exhibits tremendous scalability. According to Virtual Extension (n.d.), the number of nodes in the network has virtually no limit. Additionally, once all nodes have joined, the network exhibits very little network traffic at layers 2 and 3. All nodes listen at synchronized moments at the same frequency dictated by the pseudo-random code. If no nodes have application layer messages to transmit, no transmissions are emitted across the network. If a node detects a message, that node will repeat the received message on the next frequency hop. We can call that hop two. That node will then listen to hop three, which gives the node feedback about message receipt by other nodes. According to Virtual Extension (n.d.), battery life is also saved by each node in the VEmesh network when compared with other networks that employ popular routing techniques like Optimized Link State Routing (OLSR) and Ad Hoc Distance Vector (AODV). FHSS and Simulcast allow VEmesh nodes to save battery life by listening only at given times, each time just for a few microseconds, as opposed to continuous listening for network management traffic.

We recommend future disrupted tactical networking experiments include reactive routing protocols. In particular, we recommend observing how AODV implementations affect the network behavior during short-bursts. We do not believe that AODV is an exact fit for disrupted tactical networking in the future operating environment because it does not comply with our assumptions. AODV was designed with the assumption that an end-to-end link between sender and receiver is both possible and also maintainable for the duration of the communication session.

4. Transport Layer

TCP and UDP are the most common protocols at layer 4, but we do not believe that they are suitable given our assumptions about operating bursty networks in the future

operating environment. Like AODV and LSR at layer 3, TCP and UDP were designed with the assumption that end-to-end link with the idea of a communication session. In disruption-prone environment, an end-to-end connection may not be possible. We recommend including the DTN protocol in future experiments in disrupted tactical networking. NASA (n.d.) provides a downloadable software development kit on its website that will reduce the barrier for inclusion in future experiments. As discussed in chapter II, DTN allows each node along the route between a sender and receiver to store segments, called bundles, from the downstream node and then forwards the bundles to the upstream node when the connection with that node is possible. The question becomes, then, how nodes and clusters in disrupted tactical networks will identify up-stream and down-stream nodes when given sender and destination information. We propose approaching the answer to that question with the dual concepts of hierarchy and urgency precedence.

Hierarchy seems like an appropriate method to inform nodes and clusters how to identify up-stream and down-stream nodes when given sender and destination information. We recommend applying hierarchy by combing signal range and node mobility with the organizational chart of the units involved. To demonstrate the hierarchy concept, imagine again the amphibious raid scenario discussed at the beginning of Section C in this chapter. Recall that two Marine platoons were represented by cluster A in figure 49, rotary wing assets that inserted them are represented by cluster B, and the ARG/MEU's LFOC is cluster C. Not depicted are additional cluster Bs, comprised of various types of unmanned assets, and potentially patrol craft. We propose a hierarchical system in which the LFOC has the highest local precedence. We assign the ARG/MEU LFOC to a hierarchy level 5. Along with its placement high in the tactical scenario's organizational chart, the LFOC typically has the best available power source inherent on the amphibious ship, the strongest signal range to resident communication systems, and the greatest overall mobility. Lowest in our proposed hierarchical system is cluster A. Comprised of Marine platoons with the least mobility, cluster A's radios have the least signal range and are reliant upon battery power. We assign cluster A hierarchy level 1. The intermediate cluster Bs can fill the middle of the hierarchy. We assign hierarchy

level 2 to the unmanned assets. They too rely upon batteries to power their communications systems, but have greater mobility than the Marine platoons. Their limited dwell time makes them well-suited to serve as intermediate nodes in the disrupted tactical network. The unmanned assets may frequently return to patrol craft or the ARG/MEU to recharge. We assign patrol craft to hierarchy level 3. Patrol craft can rely on engine power for their communication platforms and they have comparable mobility to the unmanned assets. Patrol craft can dwell longer than the unmanned assets. We assign the rotary wing assets to hierarchy level 4. The rotary wing assets have greater mobility than the patrol craft although they typically have more limited dwell time. Rotary wing assets rely upon engines to power their communications platforms. Table 10 captures the scenario hierarchy.

Table 10. Scenario Example of Hierarchy

Cluster Type	**Description**	**Heirarchy Level**	**Reason**
A	Team Squad Platoon	1	Limited mobility and speed, reduced signal range, battery reliance. Low in organizational chart.
B	UAV/UGV /USV/UUV	2	Average mobility and high speed, reduced signal range, battery reliance, limited dwell time
B	Surface Vessels/ Patrol Craft	3	Increased mobility and average speed, engine power for greater signal range and greater dwell time
B	Rotary Wing Assets	4	High mobility and high speed, engine power for greater signal range, while dwell time determined by fuel
C	ARG/MEU LFOC	5	High mobility and average speed, engine power for greater signal range, more diverse communication platforms. High in organizational chart.

While hierarchy is a promising method to inform nodes of up-stream and down-stream locations, urgency precedence promises to assist the force in disciplining the EM signals in order to reduce the likelihood of detection. To demonstrate the urgency concept,

suppose that we design urgency precedence classifications as “URGENT,” “PRIORITY,” or “ROUTINE.” Messages from cluster A, intermediate clusters B, as well as from cluster C are classified with an urgency precedence based on their importance to the execution of the tactical scenario. For instance, imagine that the Marine platoons have discovered a CCIR during their raid that they need to send back to the LFOC. Suppose that reception of this CCIR by the LFOC is essential to the operation. The Marine platoon’s message is assigned “URGENT” precedence. When the “URGENT” message is transmitted from a node in the platoon, all like-nodes within signal range transmit their own messages in order to identify all nodes within their own signal range. By doing so, they leverage their geographical separation to increase the likelihood of reaching a node from a higher level in the hierarchy.

Figure 50 demonstrates network behavior during a burst intended to transmit an “URGENT” message from cluster A. The dotted lines represent the known network topology by the nodes, mapped by layer 2 and 3 message traffic at the beginning of the burst. The colored circles represent the detectable signal area, produced by the signal range of each node participating in the burst.

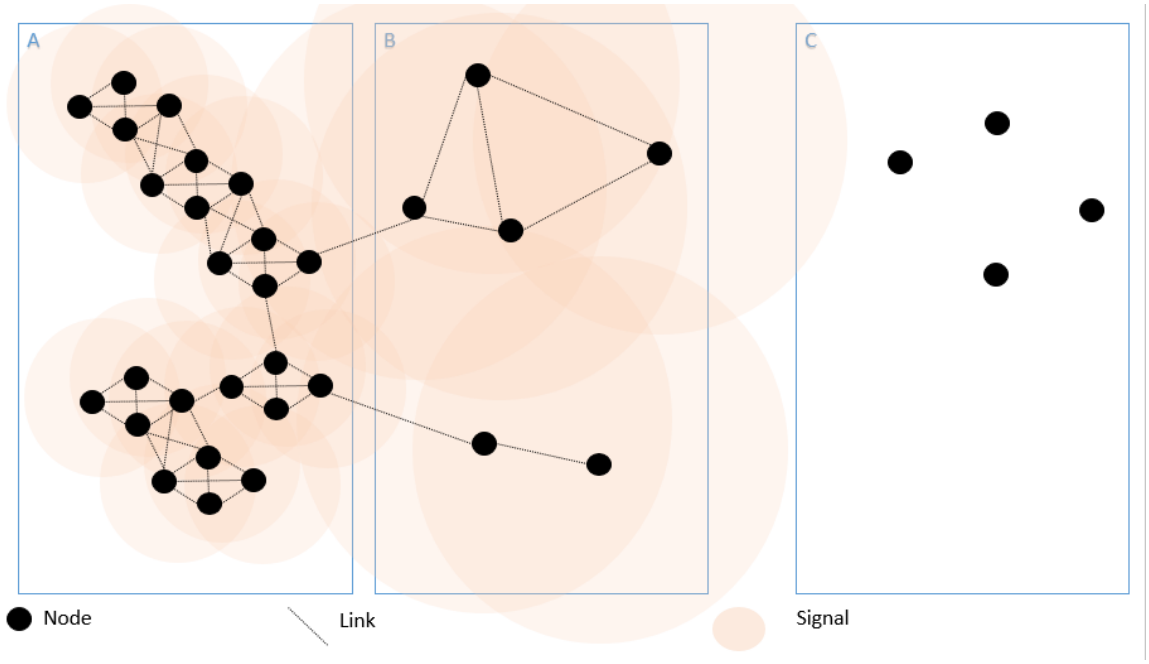

Figure 50. Network Topology During “URGENT” Message Burst

Ideally, and end-to-end link might be temporarily established between the platoon and the LFOC on ship. As depicted in Figure 50, the platoons in cluster A do not establish a link to the LFOC in cluster C. If no end-to-end link is possible, the nodes establish the best route, getting the messages as high as possible in terms of hierarchy level. The platoon's "URGENT" CCIR is transmitted across the best route, and using DTN bundle concept it is stored at a higher level in the hierarchy. The storying nodes then assume responsibility for forwarding the "URGENT" CCIR in subsequent bursts, which would occur purposefully at a different time and place in order to reduce the chance of enemy detection and targeting. As depicted in figure 50, there may be many intermediate nodes between cluster A and cluster C that end up storing the "URGENT" message. Therefore, there must be a pre-planned method to sort out duplicate messages upon receipt by cluster C. Because of the scope of the sorting task and its correlation with security mechanisms, we address this consideration in Section D, along with the rest of our CIA triad analysis.

To demonstrate the EM discipline that urgency precedence promises, consider the situation where more routine command and control information is being transmitted from the platoons to the LFOC over the horizon. Messages like location information and status reports would get the "ROUTINE" urgency precedence. When the node transmits the "ROUTINE" message during a burst, all nodes within signal range receive it. All nodes of the same hierarchy level choose not to retransmit the message. Any receiving node at a higher level in the hierarchy will store the "ROUTINE" message and assume responsibility for forwarding it during a future burst. By treating the "URGENT", "PRIORITY", and "ROUTINE" messages differently, the EM signal is disciplined. "URGENT" messages result in bigger, longer, and more frequent bursts, while "ROUTINE" messages only occur during pre-existing bursts, and are they are not repeated, which reduces the length and effective signal range of the burst. Figure 51 shows the left upper node in cluster A attempting to transmit a "ROUTINE" message. Because of its precedence, the other nodes within cluster A do not join the burst. They simply allow for any node at a higher level in the hierarchy to receive the signal directly from the transmitting cluster A node. In this way, the burst becomes more of a blip. The

burst in Figure 51 has a smaller EM signal than the burst depicted in Figure 50, and it would have a much shorter duration because no logical links are established.

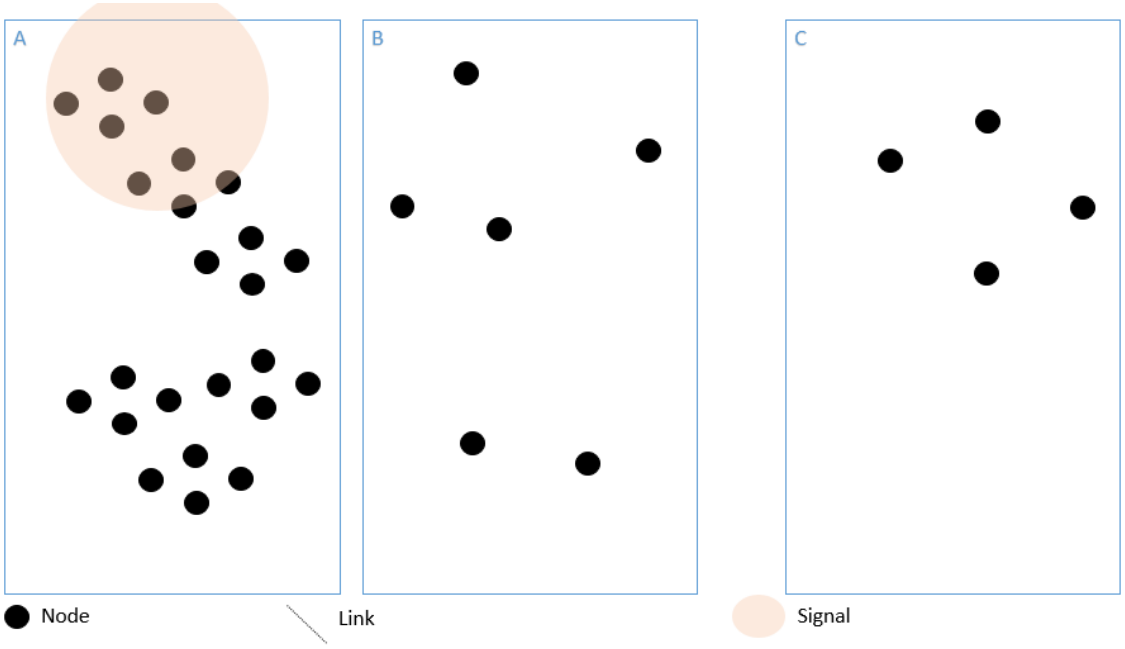

Figure 51. First Example of Network Topology during "ROUTINE" Message Burst

Figure 52 shows another example of a "ROUTINE" message burst. In Figure 52, a cluster A node successfully links with a node in cluster B. Compared with the burst in Figure 51, the burst is slightly larger and would persist for slightly longer. However, the "ROUTINE" message bursts are still less significant events than the "URGENT" burst from Figure 50.

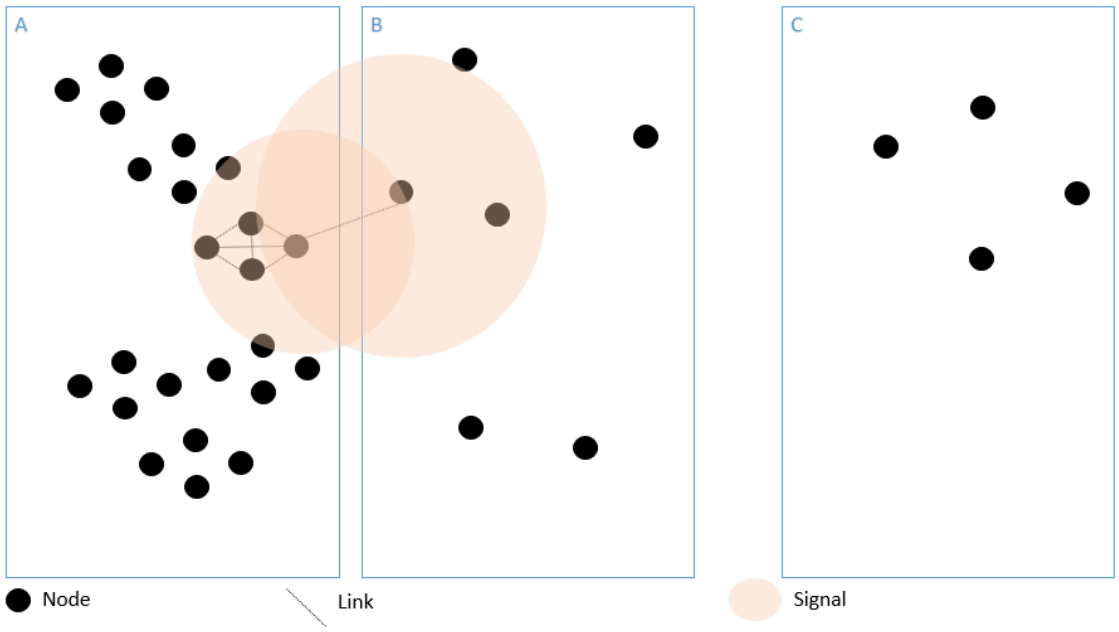

Figure 52. Second Example of Network Topology during "ROUTINE" Message Burst

5. Application Layer

The widest variety of protocols reside at the application layer. In our hypothetical disrupted tactical network in the future operating environment, a variety of protocols are likely to support command and control functions. We recommend CoT for inclusion in future experiments. CoT basic elements "what, when, and where" support a common operational picture for the force, but can also be used to help the network nodes predict the network topology. Imagine that CoT information was included in the message traffic that occurred during the bursts in Figures 50, 51, and 52. Although many cluster A nodes did not participate in the bursts, those within signal range received the transmissions. We hypothesize that the nodes can use the CoT information they received to predict what their network topology could be during the next burst. If our hypothesis is true, it may be feasible for the nodes to then exercise additional EM discipline. A node that knows that no higher-level nodes were present during a recent burst may decide not to create another burst for a "ROUTINE" message.

Our layer by layer recommendations will require a great amount of future experimentation. We were functionally constrained to use the protocols that came with our VEmesh radios during our experiments. We believe there is tremendous value in testing other protocols in similar experiments. Our future experiments will compare how the network behaves while operating with CSMA-CA with AODV to the network behavior in our experiments with FHSS with Simulcast. After that work, our future experiments will investigate how the disrupted tactical network behaves with DTN-enabled nodes. Finally, if we can overcome our functional constraints to design our own protocols, we will test our hypotheses that the hierarchy concept can give DTN-enabled nodes a means to determine upstream and downstream nodes given sender and receiver information and that the urgency concept can discipline the EM signal in the network. The next section provides a security analysis using the CIA triad.

D. SECURITY ANALYSIS

This section provides our analysis of the security considerations for tactical disrupted networks. We organize this section using the CIA triad. Although we did not use security protocols as design criteria in our experiments, we made security observations during our network behavior analysis. The recommendations in this section are preliminary, high level in nature.

As discussed in Chapter III, network managers strive to achieve balance between the elements in the CIA triad. Our discovery experiments with availability as the prime consideration. We wanted the proof of concept to show that forces can access, send, and receive information through bursts. Confidentiality and integrity were secondary goals in our study. We wanted to consider methods to support confidentiality and integrity in future disrupted tactical networks, but designing experiments to test those methods had to be left for future work. In terms of confidentiality, we wanted to consider methods to prevent an adversary from detecting, intercepting, and understanding the information. We did not include EM detection tools and an opposing force in our experiments. That is left for future work. In terms of integrity, we wanted to consider methods to ensure that information is received without being modified, whether intentionally or unintentionally.

We included authenticity in the integrity element of the CIA triad. We wanted to consider methods that the receiver can verify that the sender is a known and trusted part of the friendly force. We also wanted to consider methods to ensure that a message is not replayed, intentionally or otherwise. In Section C, we demonstrated the need to sort duplicate messages that are received through the multiple possible paths in a disrupted tactical network. Recall the Chapter III example that an "URGENT" call for fire is replayed. Without mechanisms to de-conflict multiple requests, fires could be sent to locations that friendly forces have subsequently entered. Finally, we wanted to consider methods to support non-repudiation in a disrupted tactical network. Specifically, we wanted to consider methods that the sender receives notification that a message was received by the target recipient, and secondarily that the intended receiver would be unable to deny having received the information.

Our security analysis is framed with the same underlying assumptions about disrupted tactical networks we listed in Section C. Those assumptions include that: 1) persistent signals are more likely to be detected by adversaries, 2) strict emission control, also known as *going radio silent*, will sub-optimal due to the coordination required between clusters in order to produce mission success, and 3) end-to-end connections communication links between nodes and between clusters will not be achievable or maintainable to support a communication session model.

1. Availability Analysis

Availability was of primary importance to our experiments. As a first step, we needed to prove that short-living nodes could support data transfer during a burst. In laymans terms, the network just needed to work. It is likely that availability will remain of primary importance in future work. It is well-known that availability can be impacted by the mechanisms that are intended to ensure confidentiality and integrity. Therefore, we want to seek ways to minimize the network traffic of protocols supporting confidentiality and integrity.

2. Confidentiality Analysis

This section provides a high-level analysis of achieving confidentiality while operating a disrupted tactical network. Chapter III discussed common methods to achieve confidentiality in persistent networks and why those methods are likely to perform sub-optimally in bursty networks. According to Scott and Burleigh (2007), the DTN protocol suite was designed with security in mind. The DTN protocol suite has the Bundle Security Protocol (BSP), which requires an in-depth analysis for suitability in a disrupted tactical environment. We recommend the in-depth analysis be completed in future experiments implementing DTN.

In broad terms, however, there are several available encryption schemes to support confidentiality and integrity when end-to-end communication sessions can not be supported. The first option is to create miniature sessions between each linked node. The next option is to use an asymmetric key approach like PKI. The sending node encrypts the application layer data with the target recipient's public key. A third option is to employ a symmetric key for all nodes in the friendly network. During a miniature session between nodes, key exchange or key generation is possible. However, the network traffic overhead to create sessions between links is likely to reach unacceptable levels. An asymmetric approach requires all nodes to possess all other nodes' public keys before an operation begins. The pre-distribution of keys is required because, presumably, without being able to support end-to-end links, employing a key distribution method during an operation is not supportable. The asymmetric approach requires greater node memory than other approaches, but has promise in that the network traffic overhead is negligible. The third option we considered also has negligible overhead. If all nodes in the friendly network shared a symmetric key before an operation began, no added network traffic would be required to support encryption. However, as discussed in Chapter III, distributing symmetric keys poses a challenge to network managers and confidentiality is completely lost if an adversary that discovers the symmetric key. Given our broad perspective, we believe it is necessary to further investigate the asymmetric key approach.

3. Integrity Analysis

This section provides a high-level analysis of achieving integrity while operating a disrupted tactical network. We analyze methods to verify message integrity, guard against replay, and provide the sender with notification that a message was received by the target recipient. We recommend that future work strive to prove the identity of the sender, prove that the message has not been modified, and that the message is unique. Many of the mechanisms that support confidentiality also support integrity. Therefore, recommendations from subsection 2 have bearing on this subsection. In particular, our asymmetric encryption recommendation seems well-suited to support integrity, protect against replay, and provide authentication. We recommend future work consider the use of a number once, called a *nonce*, hashes and message authentication codes that are supported by the smart use of public and private keys.

Our recommendations in this section were provided from a high-level, cursory perspective. More work is required in order to gain greater fidelity. Our intent was to begin the research conversation with a well-rounded view of the disruption-based network idea. The next chapter provides our conclusions.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND RECOMMENDATIONS

In this study, we proved the feasibility of creating our own short-living nodes. We then used the nodes in small-scale experiments and were able to successfully retrieve command and control information from a remote sensor as well as transmit movement instructions to a UGV. Although we experienced challenges with node behavior and the included materials, the nodes proved sufficient for our study of information flow and network behavior. This chapter provides our conclusions and recommendations regarding information flow and network behavior. Many of the points are summarized from our detailed analysis provided in Chapter V. We also capture some high-level conclusions regarding future node design as well as emerging manufacturing technologies.

This thesis was limited in several ways. We limited its scope by remaining purposefully in the unclassified domain. We did not use electromagnetic detection tools in order to attempt to geo-locate our experimental nodes during the duration of their interactions. That was beyond the scope of this work. We also made no attempt to modify manufacturer-set protocols in order to optimize results. The reader should keep these limitations in mind when considering the following conclusions and recommendations.

A. CONCLUSIONS

Our conclusions are organized into system-level observations, prototyping process, network behavior by layer, and security considerations.

1. System-Level Observations

The need to conduct command and control must balance with the probability that an adversary will detect and target a friendly force that is using EM signals. In figure 48, we provided a self-balancing model that roughly follows a traditional cat and mouse system. The model adequately captures both the tactical decision to employ short-living networks in a tactical situation and the impetus to develop alternative networking means such as networking by burst. Because recent adversaries have been unable to detect and target our forces by EM means, the DoD has enjoyed networking with the persistent signal model. However, planners envision a future operating environment where a signal

detected is a signal targeted. We hope that our model can serve as a starting point, improved through future research, and ultimately used by the future force.

2. Prototyping Process

We designed and created our own short-living nodes in order to study the behavior of short-living links. The process was iterative in nature. We used the Restech Norway Mini and 230 model pneumatic line throwers as surrogates for shoulder-fired grenade launchers. We compensated for the devastating launch pressures by suspending the projectiles in the barrel and by modifying production materials. Specifically, we added some structural integrity by 3D printing at greater in-fill percentages and we opted to switch to reductive manufacturing from aluminum and polyethylene stock on a CNC desktop mill. Prototype 3 was our first successful prototype, but we continued to incrementally improve its design. The result of our incremental development was Prototype 5, a simplified version containing only a rolling switch, a battery, and the VEmesh module inside a 3D-printed body. We are excited about the future of prototyping in-house, which is becoming easier with technology maturation and the burgeoning popularity of at-home manufacturing.

3. Network Behavior By Layer

The required time that a network exists is the sum of the user-driven data at the application layer, plus all subordinate layered traffic that is designed to create the network. In our experiments, the VEmesh technology provided some valuable insights. Simulcast routing removed the network requirement to transmit frequent routing messages in order to discover network nodes and maintain network topology awareness. However, the FHSS protocol suite that enabled Simulcast required that all nodes synchronize before beginning a mission. Each node needed the Gateway's timing and the pseudo-random code of frequency hops in order to participate in Simulcast. We discovered with Prototype 5, that our VEmesh nodes required an average of 13 seconds to join the network if they were powered on only after launch. We were able to reduce that time to 8 seconds when we lowered the number of possible frequencies. However, we were unable to overcome the time required by the roll-on power scheme that protects

the components on the VEmesh circuit board, in order to successfully transmit application layer traffic during the projectile's flight.

In Chapter V we provided a layer-by-layer analysis of available protocols and provide our recommendations for their inclusion in future work. These recommendations are based on three underlying assumptions about networking in the future operating environment. The three assumptions include: 1) that persistent signals are more likely to be detected by adversaries who are able to target our forces using their signals, 2) that strict radio silence will be sub-optimal because of the need to coordinate efforts in order to produce mission success, and 3) that end-to-end communication links across disparate clusters will not be feasible to maintain persistently during a single burst of the network.

4. Security Considerations

Security observations were inherent in our experiments because security is ultimately the impetus for departing from a persistent signal network. An adversary can not penetrate a network that does not exist. Likewise, an adversary must use other means to detect and target our forces if we do not emit EM signals. Considering each element in the CIA triad, we regard additional security measures as necessary to protect against the real threats of replay and other electronic attacks. Security objectives must be achieved by means that do not require a communication-session model of communication.

B. FUTURE WORK

This section makes recommendations for the next steps to develop disrupted tactical networks. This study focused on the creation of short-living nodes. We believe future work should focus on short-living links. The projectiles themselves were just one type of node. A network of short-living nodes will feature multi-domain nodes. Some nodes will be ground-based, others will be surface, sub-surface, aerial, and others may even have an orbital nature.

Future work that is immediately pursuable is network behavior study and analysis of diverse protocols inherent in commercially-available mesh radios. GoTenna (n.d.) recently released a mesh version of their products that are designed for use in areas not

serviced by cellular companies. Persistent Systems' Wave Relay radios provide another system that is readily available for immediate testing. We recommend comparing the network behavior of assets that use CSMA-CA at layer 2 and AODV at layer 3 with our VEmesh TDMA, FHSS, and Simulcast observations.

We also recommend future tests with DTN-enabled nodes. We believe DTN is a good fit given our assumption that end-to-end connections will not be possible in the future operating environment. We recommend developing protocols to give the DTN-enabled nodes a method to determine whether a sender is up-stream and down-stream relative to the target recipient of a given message. We suggest a hierarchical method based on the organizational hierarchy and relative mobility of the nodes. We also suggest applying an urgency precedence in order to exercise greater EM discipline in the network. An "URGENT" message can produce a different network behavior than the behavior produced by a "ROUTINE" message if designed into the protocol.

Finally, we recommend incorporating EM detection and geolocation tools into future experimentation. By designing future experiments to include an opposing force with real EM detection and geolocation capabilities, which would provide insight about the allowable duration of a single burst and present the opportunity to further refine our proposed systems-theory model for operating short-living networks.

APPENDIX A. ARDUINO TIMER SKETCH

Appendix A is the very basic Arduino Sketch that drove Prototype 1.

```
/*  Timed Parachute Release
*  Kline
*/

#include <Servo.h>

Servo myservo;          // create servo object to control a servo
                        // twelve servo objects can be created on most boards

void setup() {
   delay(2000);}

void loop() {
   myservo.attach(3);   //connect servo on pin 3
   myservo.write(0);    //tell servo to begin at pos 0
   delay(500);
   myservo.write(90);   // tell servo to go to position 90
   delay(500);          // waits 15ms for the servo to reach the position
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. PROTOTYPE 3 SKETCH

Appendix B is the Arduino Sketch driving Prototype 3. This sketch was designed by Naval Postgraduate School Researcer, Eugene Bourakov.

```
/-----------------------------------------------
//---CENETIX VE Projectile payload ----
//-----------------------------------------------
#include <SoftwareSerial.h>
#include <Wire.h>
#include <SPI.h>
#include <Servo.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
//#include <Adafruit_GFX.h>
//#include <Adafruit_SSD1306.h>

// software serial #1: RX = digital pin 8, TX = digital pin 9
SoftwareSerial radioPort(8, 9);

Servo myservo;              // Calls Servo Library

int STATUS_LED = 13;
int TILTBALL_PIN = 7;
const int axisXpin = A3;
long cntRst=0;
float baseAlt=0;
const int numReadings = 5;  // for moving average filtering
int ma[numReadings];        // moving average array
boolean LAUNCHED=false;
boolean DEPLOYED=false;
boolean TOUCHDOWN=false;
boolean INITIALIZED=false;
boolean TRANSMIT=true;
float prevPressure=0;
//float prevprevPressure=0;
//long prevMillis=0;
int accelerationThreshold=700;
int afterLaunchDelay=90;  // delay at least 1 second to strat sensing tilt ball
float speedOfPressureChange=0;

Adafruit_BMP280 bme; // I2C
//Adafruit_SSD1306 SSDdisplay;
//-----------------------------------------------
void setup()
{
   Serial.begin(9600);
   radioPort.begin(9600);
   pinMode(STATUS_LED, OUTPUT);
   pinMode(TILTBALL_PIN, INPUT);
```

```
   if (!bme.begin()) {
     Serial.println(F("Could not find a valid BMP280 sensor, check wiring!"));
     while (1);
   }

   Serial.println("Projectile payload ready");
     //SSDdisplay.begin(SSD1306_SWITCHCAPVCC, 0x3C);  // initialize with the I2C addr 0x3C
(for the 128x64)
   //SSDdisplay.display();
   //displayInit();
   //delay(1000);

}
//-----------------------------------------------------
void loop() {
      String inData="";
       while (radioPort.available() > 0) {
       char inByte = radioPort.read(); // Receive a single character from the software serial port
       inData.concat(inByte);        // Add the received character to the receive buffer
       if (inByte == 13)
       {
           parseInData(inData);
           inData = "";
       }
   }

   if(cntRst%5 ==0){
     // check speed every 100 ms
     speedOfPressureChange=getSpeedOfPressureChange();
     if(speedOfPressureChange>0.0)    Serial.println(speedOfPressureChange);
   }

   //if(!LAUNCHED && abs(analogRead(axisXpin))>accelerationThreshold ){
   if(INITIALIZED        &&        !LAUNCHED        &&        (speedOfPressureChange>1.5        ||
abs(analogRead(axisXpin))>accelerationThreshold) ){
     Serial.println("launched");
     LAUNCHED=true;
     cntRst=0;
   }

   if    (digitalRead(TILTBALL_PIN)    ==    HIGH    &&    LAUNCHED    &&    !DEPLOYED    &&
cntRst>afterLaunchDelay) {
     deployParachute();
   }

   if(LAUNCHED && DEPLOYED &&  TRANSMIT && cntRst>100){
     // continue to transmit for a second to send deployment status
     TRANSMIT=false;
     cntRst=0;
   }

   if(LAUNCHED     &&     DEPLOYED     &&          !TRANSMIT     &&     cntRst>100     &&
abs(analogRead(axisXpin))<accelerationThreshold/3){
     Serial.println(analogRead(axisXpin));
     Serial.println("touchdown");
```

```
      TOUCHDOWN=true;
      TRANSMIT=true;
      //cntRst=0;
    }

    updateGCS();

    cntRst+=1;

    if(DEPLOYED && cntRst>6000) {
      // re-initialize in about 30 seconds after deployment
       initPayload();
    }

    if(!INITIALIZED && cntRst>200) {
      // after about a second initialize altitude measurements to zerose AGL
      blinkLED(STATUS_LED);
      INITIALIZED=true;
      initPayload();
    }

    //  main cycle takes about 20ms
    //delay(1);
}
//-----------------------------------------------------
void updateGCS() {

    char buf[100];
    int
payloadStatus=getBooleanInt(LAUNCHED)+2*getBooleanInt(DEPLOYED)+4*getBooleanInt(TO
UCHDOWN);
    String myString="1 "+String(millis())+" "+String(int(getAlt()-baseAlt))+" "+String(payloadStatus);
    //Serial.println(myString);
    myString.toCharArray(buf,myString.length()+1);
    if(TRANSMIT) {
      //Serial.println(myString);
      radioPortSendOut(buf);
    }
}
//-----------------------------------------------------
int getBooleanInt(boolean val){
  return  val ? 1  : 0;
}
//-----------------------------------------------------
float getSpeedOfPressureChange() {
  //int h=millis()-prevMillis;
 int  h=50;

  // get derivative with  dy/dx=(y(x+h)-y(x))/h

  float deltaPressure=abs(bme.readPressure()-prevPressure);
  prevPressure=bme.readPressure();
  // or dy/dx=(y(x+h)-y(x-h))/2h
  // will try later
```

```
  return deltaPressure/h;

  // or with 3-points method
  // y(1)'=(y(-1)-4*y(0)+3*y(1))/2*h
  //float deltaPressure=prevprevPressure-4*prevPressure+3*bme.readPressure();
  //float deltaPressure=-prevprevPressure-0*prevPressure+bme.readPressure();
  //prevprevPressure=prevPressure;
  //prevPressure=bme.readPressure();
  //return deltaPressure/2*h;

}
//---------------------------------------------------
float getAlt() {
    float alt=bme.readAltitude(1031.00);
    int i;
    for(i=1;i<numReadings;i++){
      ma[i-1]=ma[i];
    }
    ma[numReadings-1]=alt;
    float sum=0;
    for(i=0;i<numReadings;i++){
      sum+=ma[i];
    }
    alt=sum/numReadings;

    return alt;
}
//---------------------------------------------------
void radioPortSendOut(char* buf) {
    radioPort.write(buf);
    radioPort.write(13);
    //blinkLED(STATUS_LED);
}
//---------------------------------------------------
void blinkLED(int pin) {
    digitalWrite(pin, HIGH);   // turn the LED on (HIGH is the voltage level)
    delay(10);              // wait for a second
    digitalWrite(pin, LOW);    // turn the LED off by making the voltage LOW
}
//---------------------------------------------------
void parseInData(String inData) {
    String header="";
    String cmd="";
    Serial.println(inData);
    if(inData.indexOf("#")>0){
      inData.toUpperCase();
      header=inData.substring(0,inData.indexOf("#"));
      cmd=inData.substring(inData.indexOf("#")+1);
      Serial.println(header+"  "+cmd);

      if(cmd.startsWith("INIT")) {
        initPayload();
      }

      if(cmd.startsWith("DEPLOY")) {
```

```
        deployParachute();
      }
    }

}

//-----------------------------------------------------
void initPayload() {
    LAUNCHED=false;
    DEPLOYED=false;
    TOUCHDOWN=false;
    baseAlt=getAlt();
    cntRst=0;
    TRANSMIT=true;
    servoControl(90);  //set to init position
    Serial.println("initialized");
}
//-----------------------------------------------------
void deployParachute() {
    LAUNCHED=true;
    DEPLOYED=true;
    updateGCS();
    servoControl(0);
    cntRst=0;
    Serial.println("parachute deployed");
}
//-----------------------------------------------------
void servoControl(int pos){
    myservo.attach(6);  //connect servo on pin 6
    delay(15);
    myservo.write(pos);
    delay(400);
    myservo.detach();
}
/*
//-----------------------------------------------------

void displayInit() {
  SSDdisplay.clearDisplay();
  SSDdisplay.setTextSize(2);
  SSDdisplay.setTextColor(WHITE);
  SSDdisplay.setCursor(0,8);
  SSDdisplay.println("  CENETIX");
  SSDdisplay.display();
}
//-----------------------------------------------------

void displayData() {
  SSDdisplay.clearDisplay();
  SSDdisplay.setTextSize(1);
  //SSDdisplay.setTextColor(WHITE);
  SSDdisplay.setCursor(0,0);
  SSDdisplay.println("   CNTX Projectile");
  SSDdisplay.setCursor(0,10);
  if(INITIALIZED) {
```

```
    SSDdisplay.println("AGL: "+String(getAlt()-baseAlt)+" meters");
  } else {
    SSDdisplay.println("Alt: "+String(getAlt()-baseAlt)+" meters");
  }
  SSDdisplay.setCursor(0,20);
  SSDdisplay.println("Tmp: "+String(float(bme.readTemperature())-4.0)+" C");
  SSDdisplay.display();
}
*/
//-------------------------------------------------
```

APPENDIX C. PHASE III EXPERIMENT

Appendix C is extracted from the CENETIX campaign of experiments in fiscal year 2017. The below table summarizes the experiments Phase III of this thesis. The table was originally provided to all CENETIX participants for informational use. After the experiment, participants were provided a copy with the final results and analysis in text below the table.

CENETIX EXPERIMENTS Appendix I (Part C, Phase II) to Annex C

Short Title	**Projectile Network Testing—Camp Roberts, CA**
Phase	**Part A, Phase II (21 March)**
Experiment Objectives	Conduct command and control through short-living nodes.
Tactical Level Problem	Current networking framework, that of persistent connection, is not suitable to operate under electromagnetically hostile conditions, especially when adversary can geo-locate.
Research Questions	*1. Can critical information be transmitted using short-living networking nodes?* *2. What behavior is exhibited by layer of the OSI stack during transmission?* *3. How might projectiles be integrated into future scenarios of short-living networks?*
Technical Objectives	1. Examine network behavior during projectile flight.
Partner Interest Area	n/a
Integration Variables	tbd
Reachback Model	n/a
Constraints	1. Number of Nodes (3) 2. Flight Time—PLT limits (10s) 3. Internal components (VEMesh, Arduino)
Criteria	1. UGV communication
Location	**Camp Roberts, CA**
Date	**Tue, 21 March**
Players	1. Monitoring and Control team (NPS) 2. Camp Roberts range support
MIO-CWMD Testbed Infrastructure	• CENETIX Testbed Portal • Deployable local MANET components • NPS SA and data capture tools

Local Test Bed Components in Use	• Wireless mesh network • Observer Notepad—not used • Projectile launcher –large PLT		
Scenario w/MSEL	n/a		
Scheme	*1. Prepare the gateway, a projectile, and a UGV.* *2. Launch projectile.* *3. Transmit C2 signal from projectile to the UGV*		

	Activity	**NPS (PST)**	**DC (+3)**
Phase Sequence	Bench test projectile, gateway, remote node functionality	1430-1500	1730-1800
	Shoot projectile, testing operation (repeat as needed)	1500-1700	1800-2000
	Shoot projectile, convey command to UGV	"	"
	Recover equipment	1700-1730	2000-2030
	Hotwash	1730-1800	1730-1800

RQ 1	*1. Can critical information be transmitted using short-living networking nodes?*	
	MoPs	**Data Collector**
	a) Is data received at the remote node? How much?	ENGR
	b) Is data received at the gateway? How much?	ENGR
	c) What is the flight time?	ENGR
	d) Length of time link is closed?	ENGR
RQ 2	*2. What behavior is exhibited by layer of the OSI stack during transmission?*	
	MoPs	**Data Collector**
	a) Collect all communications by layer using sniffer for analysis.	MEJIA
RQ 3	*3. How might projectiles be integrated into future scenarios of short-living networks?*	
	MoPs	**Data Collector**
	a) What are the characteristics of flight of the projectile?	KLINE
	b) What is the signal range?	KLINE
	c) What refinements can be made for future experiments?	PI
	d)	

Other Data Collection	**Network Logs**	• System Latency	
	Tech Obsns	• Network S/W issues • Network H/W issues	Bourakov Bourakov
	Obsr Notepad	• Text chat thread	Wendt
	SA View	• Screen captures of SA View COP	Raap
Observer Notepad Naming Convention	**Callsign**	"PI" - Bordetsky "NOCRear" - x (at NPS) "TOC" - Wendt "OPS" - Mullins "ENG" - Bourakov	"Big-Un" - Kline
Team Assignments	**None**		

Notes: Ground test (no launch) successful.

6 launch tests:

1st: Battery fault, no power after shot. Parachute did not deploy. Since Arduino did not detect conditions for deployment, the projectile never sent C2 information to UGV.

2nd: Parachute did not deploy. Attempting to eliminate reasons to include: accelerometer malfunction & barometer malfunction (program logic requires either/or acceleration sensor or 10m altitude change to activate tilt sensor, which deploys parachute and begins projectile node communication), or reset sensor triggered during flight, or battery failure.

3rd-5th: UGV conducted maneuvers, indicating that the data was received by the remote node. However, the projectile's parachute did not deploy.

Analysis: Parachute involved for 2 reasons: increase flight time and protect node. Node has proven to withstand impact on grass landings, and flight time without parachute is sufficient for messaging. Future prototype will eliminate mechanisms for parachute (accelerometer, barometer, tilt sensor, parachute, plunger, springs, servo, and lever).

Additionally, multiple impact revealed that the projectile shaft (came with PLT equipment) unscrews and has a cavity inside suitable for insertion of future prototype. Previously, the projectile shaft appeared as a single unit, its cap glued/fused to the body. In PLT3 prototype, we used it simply to absorb the impact of the launcher, push the projectile up, then separate.

Way Ahead: Determine program logic in new prototype. Develop new prototype for internal to the shaft.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Air Command Water Rockets (n.d.). The shadow—build log. Retrieved November 13, 2016 from http://www.aircommandrockets.com/shadow.htm

Alberts, D. Hayes, R. (2002). Code of best practice for experimentation. command and control research program. Washington, DC, Retrieved from www.dodccrp.org.

Alberts, D. Hayes, R. (2005). *Campaigns of experimentation* [Ebrary version]. Retrieved from http://internationalc2institute.org/ccrp-books.

Angevine, R. G. (2011). Hiding in plain sight: The U.S. Navy and dispersed operations under EMCON, 1956–1972. *Naval War College Review*. Spring:79-95. Retrieved April 16, 2016 from https://www.usnwc.edu/getattachment/bfd7502d-682c-444d-946c-63245227ae68

Arduino. (n.d.) What is Arduino? Retrieved from https://www.arduino.cc/en/Guide/Introduction

Barabási, A. (2010). *Bursts.* New York, NY: Penguin Group.

Barabási, A. (2014). *Linked: The new science of networks.* New York, NY: Perseus Publishing.

Barker, E. (2016). *Guideline for using cryptographic standards in the federal government: Cryptographic mechanisms* (NIST Special Publication 800–177B). Retrieved from http://dx.doi.org/10.6028/NIST.SP.800-175B

Bordetsky, A., & Netzer, D. (2010). Testbed for tactical networking and collaboration. *The International C2 Journal,* 4(3). Retrieved from http://www.dodccrp.org/files/IC2J_v4n3_B_Bordetsky.pdf

Bordetsky, A. (2012*). Patterns of tactical networking services. Chapter 18 of Cloud Computing Service and Deployment Models,* IGI Global.

Bordetsky, A., Benson, S., and Hughes, W. (2016). *Hiding comms in plain sight*. Retrieved August 14, 2017 from https://www.afcea.org/content/Article-hiding-comms-plain-sight

Capra, F. (1996). *The Web of Life: A new scientific understanding of living systems*. New York: Anchor Books.

Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., Weiss, H. (2007). "Delay-Tolerant Networking Architecture (RFC 4838)." The Internet Engineering Task Force [IETF] Trust. Available at: https://tools.ietf.org/html/rfc4838

Comer, D. (2014). *Computer Networks and Internets*, 6th Edition. Upper Saddle River, NJ: Pearson/Prentice Hall.

Cursor On Target Office. (2013). Cursor on target 101 briefing. Retrieved July 30, 2017 from www.dtic.mil/dtic/tr/fulltext/u2/a578867.pdf

Deblois, B.M. (1998). Space sanctuary: A viable national strategy. *Airpower Journal* [Winter, 1998]. Retrieved October 31, 2016 from http://www.au.af.mil/au/afri/aspj/airchronicles/apj/apj98/win98/deblois.html

Department of the Army, (2004). *Multiservice tactics, techniques, and procedures for nuclear, biological, and chemical reconnaissance* . Retrieved May 18, 2017 from http://armypubs.army.mil/doctrine/DR_pubs/dr_a/pdf/atp3_11x37.pdf

Department of the Navy. (2013). *Information dominance roadmap, 2013–2028.* Washington, DC: Author.

Department of the Navy. (2015). *A cooperative strategy for the 21st century seapower.* Washington, DC: Author.

Dortch, M. (2015). Notice of apparent liability for forfeiture. Federal Communication Commission. Retrieved May 18, 2016 from http://transition.fcc.gov/Daily_Releases/Daily_Business/2013/db0802/FCC-13-106A1.pdf

Federal Information Security Management Act of 2002. H.R. 3844 — 107th Congress. Retrieved from https://www.govtrack.us/congress/bills/107/hr3844

Granovetter, M. (May 1973). The strength of weak ties. *American Journal of Sociology*, Vol 78, Issue 6. 1360–1380. Retrieved October 15, 2016 from https://sociology.stanford.edu/sites/default/files/publications/the_strength_of_weak_ties_and_exch_w-gans.pdf

Gruss, M. (14 October, 2013). Companies see market for systems to counter GPS jamming devices [web document]. Space News. Retrieved May 16, 2016 from http://spacenews.com/37706companies-see-market-for-systems-to-counter-gps-jamming-devices/

International Organization for Standardization (ISO). (1996). Information technology—Open systems interconnection—Basic reference model: The basic model. *International Standard ISO/IEC 7498–1*. Retrieved from http://standards.iso.org/ittf/PubliclyAvailableStandards/s025022_ISO_IEC_7498-3_1997(E).zip

Joint Chiefs of Staff. (2010). *Department of Defense dictionary of military and associated terms* (Joint Publication 3-0). Washington, DC: U.S. Government Printing Office.

Joint Chiefs of Staff. (2011a). *Joint operations* (Joint Publication 3-0). Washington, DC: U.S. Government Printing Office.

Joint Chiefs of Staff. (2011b). *Command and control for joint maritime operations* (Joint Publication 3-32). Washington, DC: U.S. Government Printing Office.

Joint Chiefs of Staff. (2013). *Command and control for joint maritime operations* (Joint Publication 3-32). Washington DC: U.S. Government Printing Office.

Kan, S. (2007). *China's anti-satellite weapon test* (CRS Report No. RL22652). Retrieved from Congressional Research Service website: http://www.fas.org/sgp/crs/intel/RL22652.pdf

Kristan, M., Hamalainen, J., Robbins, D., & Newell, P. (2009).Cursor-on-target message router user's guide. Mitre Corporation, Bedford, MA. Retrieved from https://www.mitre.org/sites/default/files/pdf/09_4937.pdf

Lewis, J. (2014, August 9) They shoot down satellites, don't they? *Foreign Policy*. Retrieved from http://foreignpolicy.com/2014/08/09/they-shoot-satellites-dont-they/

Misra, S. C., Misra, S., & Woungang, I. (Eds.). (2009). *Guide to wireless mesh networks*. London, England: Springer. Retrieved March 12, 2017 from http://libproxy.nps.edu/login?url=http://link.springer.com/book/10.1007/978-1-84800-909-7/page/1

National Aeronautics and Space Administration . (n.d.). Disruption tolerant networking. Retrieved from https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt_dtn.html

Patwari, N., Ash, J., Kyperountas, S., Hero III, A., Moses, R., and Correal, N. (July 2005). Locating the nodes. *IEEE Signal Processing Magazine.* Volume 22, Issue 4. DOI: 10.1109/MSP.2005.1458275

Rafael. (n.d.) Firefly: Ground assault tactical intelligence. Rafael Armament Development Authority, Ltd. Missile Division. Retrieved October 18, 2016 from http://www.rafael.co.il/marketing/SIP_STORAGE/FILES/0/730.pdf

Restech Norway. (n.d.). Restech Norway. Available at: http://restech.no/

Sayed, A., Tagrihat, A., & Khajehnouri, N. (2005). Network-based wireless location. *IEEE Signal Processing Magazine.* Volume 22, Issue 4. DOI: 10.1109/MSP.2005.1458275

Scott, K., & Burleigh, S. (2007) "Bundle protocol specification (RFC 5050)," The IETF Trust. Retrieved May 15, 2017 from https://tools.ietf.org/html/rfc5050

Sehl, T. (2013). The Viability of a DTN system for concurrent military application Master's thesis, Naval Postgraduate School, Monterey, CA.

Senge, P. M. (2006). *The Fifth Discipline*. New York: Doubleday.

Singer, P. (2009). *Wired for War*. New York: The Penguin Press.

Socolofsky, T., & Kale, C. (1991). "A TCP/IP Tutorial (RFC 1180)." The Internet Engineering Task Force [IETF] Trust. Available at: https://tools.ietf.org/html/rfc1180

Stanley, W., & Jeffords, J. (2006). *Electronic Communications: Principles -and Systems, 1st Edition*. Thompson Delmar Learning. Cliffton Park, NY.

Statnikov, R., & Statnikov, A. (2011). *The Parameter Space Investigation Method Toolkit*. Artech House, Boston.

Statnikov, R., & Matusov, J. (1995). *Multicriteria Optimization and Engineering*. Springer Science & Business Media.

Sterman, J.D. (2010). *Business Dynamics: Systems Thinking and Modeling for a Complex World.*New York: McGraw-Hill.

Stoneburner, G. (2001). Underlying Technical Models for Information Technology Security: Recommendations of the National Institute of Standards and Technology [NIST SP 800–33]. Government Printing Office. Washington, DC,

Subramanian, M. (2010). *Network management: principles and practice.* Safaribooksonline.com. Pearson Education India. Retrieved from http://techbus.safaribooksonline.com/book/networking/network-management/9788131727591

United States Marine Corps. (2016a). *Marine Corps operating concept.* Washington, DC: Author.

United States Marine Corps. (2016b). *Marine Corps operating concept brochure.* Washington, DC: Author.

Virtual Extension. (n.d.). Virtual extension: Products. Virtual Extension. Retrieved from http://www.virtual-extension.com/products/technology/

Wang, J., Xie, B., & Agrawal, D. (2009) *Journey from mobile ad hoc networks to wireless mesh network.s* In S. Misra, S.C. Misra & I. Woungang (Eds.), *Guide to Wireless Mesh Networks* (pp. 1–29). Springer-Verlag London. DOI: 10.1007/978-1-84800-909-7

Weiner (1961). *Cybernetics: or, Control and Communication in the Animal and the Machine* [2d Edition]. M.I.T. Press. New York, NY.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California